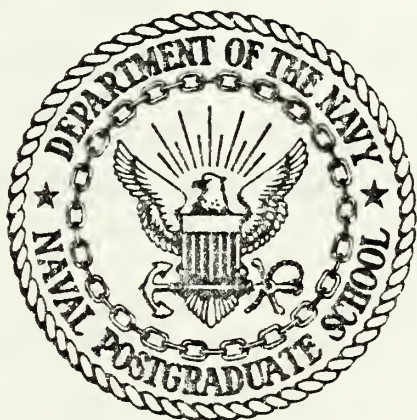


100L
CALIFORNIA 95943-8002

NAVAL POSTGRADUATE SCHOOL

Monterey, California



65743

THESIS

UTILIZATION OF NUMERICAL OPTIMIZATION
TECHNIQUES IN THE DESIGN OF ROBUST
MULTI-INPUT MULTI-OUTPUT CONTROL SYSTEMS

by

Vernon Curtis Gordon

September 1984

Thesis Advisor:

D. J. Collins

Approved for public release; distribution unlimited.

T238935

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Utilization of Numerical Optimization Techniques in the Design of Robust Multi-Input, Multi-Output Control Systems		5. TYPE OF REPORT & PERIOD COVERED Doctor of Philosophy September 1984
7. AUTHOR(s) Vernon Curtis Gordon		6. PERFORMING ORG. REPORT NUMBER
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Postgraduate School Monterey, California 93943		8. CONTRACT OR GRANT NUMBER(s)
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Postgraduate School Monterey, California 93943		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		12. REPORT DATE September 1984
		13. NUMBER OF PAGES 175
		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report) Approved for public release; distribution unlimited.		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) Multi-input Multi-output Control, Robustness, Modern Control, Multivariable Systems.		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) A direct design method for solving the problem of robustness to cross-coupling perturbations in multivariable control systems is presented. The method uses numerical optimization procedures to manipulate the system feedback gains as direct design variables. The manipulation is accomplished in a manner that produces desired performance by pole placement and robustness by modification of the minimum singular values of the system return difference matrix.		

DD FORM 1 JAN 73 1473

EDITION OF 1 NOV 65 IS OBSOLETE

S/N 0102-LF-014-6601

UNCLASSIFIED

1 SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

Channels affected by cross-coupling perturbation may be recognized by the character of their transfer function plot. The mechanism used by the pole placement and robustness routine in obtaining a robust design is evident from the gain changes associated with the transfer function diagram and the zero shifts shown on pole-zero plots. The pole placement and robustness routine uses gain equalization and zero assignment to modify the characteristics of the system in the areas of low singular values, producing a robust design.

A modification of the pole placement and robustness routine that may be applied to the design of robust observers is also presented. Using feedback and filter gains as direct design variables a practical design procedure for robustness recovery in observer based systems is obtained.

Approved for public release; distribution unlimited.

Utilization of Numerical Optimization Techniques
in the Design of
Robust Multi-input Multi-output Control Systems

by

Vernon C. Gordon
Commander, United States Navy
B.A.E., Auburn University, 1968
M.S.A.E., Naval Postgraduate School, 1970

Submitted in partial fulfillment of the
requirements for the degree of

DOCTOR OF PHILOSOPHY

ABSTRACT

A direct design method for solving the problem of robustness to cross-coupling perturbations in multivariable control systems is presented. The method uses numerical optimization procedures to manipulate the system feedback gains as direct design variables. The manipulation is accomplished in a manner that produces desired performance by pole placement and robustness by modification of the minimum singular values of the system return difference matrix.

Channels affected by cross-coupling perturbation may be recognized by the character of their transfer function plot. The mechanism used by the pole placement and robustness routine in obtaining a robust design is evident from the gain changes associated with the transfer function diagram and the zero shifts shown on pole-zero plots. The pole placement and robustness routine uses gain equalization and zero assignment to modify the characteristics of the system in the areas of low singular values, producing a robust design.

A modification of the pole placement and robustness routine that may be applied to the design of robust observers is also presented. Using feedback and filter gains as direct design variables a practical design procedure for robustness recovery in observer based systems is obtained.

TABLE OF CONTENTS

I.	INTRODUCTION	12
II.	SINGLE-INPUT SINGLE-OUTPUT SYSTEMS	14
III.	MULTIVARIABLE SYSTEMS	26
IV.	OPTIMIZATION	33
V.	OPTIMIZATION DESIGN PROCEDURE	52
VI.	INTRODUCTORY PROBLEM	60
VII.	A HELICOPTER STABILITY PROBLEM	82
VIII.	SIMPLE OBSERVER	113
IX.	RCBUST OBSERVER DESIGN	124
X.	CCNCLUSIONS	130
	LIST OF REFERENCES	173
	INITIAL DISIRIBUTION LIST	175

LIST OF TABLES

1.	Comparative Results Simple Problem	77
2.	CH-46 Helicopter Parameter Definitions	84
3.	Design One Pole Placement	92
4.	Helicopter Problem Feedback Gains	104
5.	Observer Parameter Data	116

LIST OF FIGURES

2.1	Basic Gain and Phase Margin Determination	
	Model	15
2.2	Classical Bode Plot	16
2.3	Nyquist Plot of Stable System	17
2.4	Additively Perturbed System	18
2.5	Additive Nyquist Plot	19
2.6	Nyquist for Inequality Additive Condition	20
2.7	Multiplicative System	21
2.8	Nyquist Plot for Multiplicative System	22
2.9	Typical Equivalent Feedback System	23
2.10	Polar Plot of an Optimal System	24
3.1	Nyquist D Contour	28
3.2	Basic Multi-input, Multi-output System	29
3.3	Additive Perturbation	30
3.4	Multiplicative Perturbation	32
4.1	Design Space for Example Problem	34
4.2	Design Space for Column Problem	36
4.3	Illustrative Example for Iteration	38
4.4	Plot of Various Points With Zero Gradient	40
4.5	Constrained Optimization Example	42
4.6	Organization of ADS Program	44
4.7	Golden Section Diagram	45
4.8	Steepest Descent Algorithm	48
4.9	Fletcher-Reeves Conjugate Direction	
	Algorithm	49
5.1	Universal Gain and Phase Singular Value Plot	55
5.2	Observer Implementation	58
6.1	Basic Multi-input Multi-output System	61

6.2	Multivariable Nyquist for $2s+3/(s+1)^2$	62
6.3	Minimum Singular Value Plot, Example Problem	64
6.4	Nyquist Diagram of $1/s+1$	65
6.5	Perturbed System	66
6.6	Singular Value Plot for Simple Problem	67
6.7	Closed-loop Poles for Pole Placement Only Case	68
6.8	Singular Value Plot Case Three	70
6.9	Open-loop Transfer Function 2-1 for Baseline	72
6.10	Transfer Function 2-1 Optimized(3 Var)	73
6.11	Transfer Function 1-1 Baseline	74
6.12	Transfer Function 1-1 Optimized (3 variables)	75
6.13	Closed-loop Pole-Zero Plot Case 3	76
6.14	Pole-Zero Plot Simple Problem (4 Variables)	79
6.15	System Block Diagram	80
7.1	Feedback Control Structure	85
7.2	System Diagram	86
7.3	Alphatech Design Singular Value Plots	88
7.4	Design One Perturbation Input	89
7.5	Design Two Perturbation Input	90
7.6	Transfer Function $\delta_B - \delta_B$ Design One, Nonoptimized	93
7.7	Transfer Function $\delta_B - \delta_B$ Design One , Optimized	94
7.8	Transfer Function $\delta_C - \delta_B$ Design One, Nonoptimized	96
7.9	Transfer Function $\delta_C - \delta_B$ Design One, Optimized	97
7.10	Pole-Zero Plots for Design 1	98
7.11	Pole-Zero Plots for Design 1 (cont.)	99
7.12	Pole-Zero Plots for Design 1 (cont.)	100
7.13	Pole-Zero Plots for Design 1 (cont.)	101

7.14	The δ_B to v Frequency Response, Nonopt. . . .	102
7.15	The δ_B to v Frequency Response, Optimized . .	103
7.16	Transfer Function $\delta_c - \delta_B$ Design 1 Case2, Nonopt.	105
7.17	Transfer Function $\delta_c - \delta_B$ Design 1 Case 2, Opt .	106
7.18	Transfer Function $\delta_c - \delta_B$ Design Two, Nonopt. .	108
7.19	Transfer Function $\delta_c - \delta_B$ Design Two, Optimized	109
7.20	Singular Value Plot Design Two	110
7.21	Time Response Design Two	111
8.1	Simple Observer	113
8.2	Nyquist Plot	115
8.3	Nyquist Plot for Robustness Recovery	117
8.4	Singular Values of Observer System	118
8.5	Singular Value Comparison Plot	120
8.6	Nyquist for Computed Robustness Recovery . . .	121
8.7	Time Response Plot for Simple Observer	122
9.1	Observer Based Controller	125
9.2	Singular Value Plot of Observer Results . . .	126
9.3	Time Response for Observer System	127

ACKNOWLEDGEMENTS

For over fifteen years the United States Navy has allowed me to make practical application of the science of flight as a Naval Flight Officer and as a flight test project officer. I wish to express my appreciation to the Navy and the citizens of the United States for the opportunity to further expand my knowledge of aviation by pursuing this degree.

I would also like to express a sincere thank you to the Doctoral Committee of professors Platzner, Gawain, Latta, Strum and Collins for their support in these efforts. Of course special thanks must go to Professor D.J. Collins for heading the committee and having worked so long and hard with me on two theses. Without his support this effort would never have been completed.

Thanks are also expressed to Professor G.N. Vanderplaats for allowing me to use the Automated Design Synthesis code. His help in integrating the code into the pole placement and robustness design program and the insight into the power of numerical optimization that he provided were extremely beneficial to the outcome of this thesis.

The Naval Postgraduate School Library staff and Roger Martin are appreciated tremendously for providing excellent service. No request for reference material was ever too difficult for them to handle. The W.R. Church Computer Center is acknowledged for their support in the computational processing for this thesis. I would also like to acknowledge the work of CDR Al Diel and LTs Cliff Cooksey, John Hoden and Mike Iaptas for the work done to provide an excellent control system analysis package at the Naval Postgraduate School. Their efforts made the analysis of data from the pole placement and robustness routine much easier.

I would also like to thank my wife, Janice, and children, Bradley and Melissa. As anyone knows who has undertaken an effort of this magnitude without their love and support this would have been impossible. They have "gone it alone" quite often so that this paper could be completed.

Above all thanks to Him who started it all.

I. INTRODUCTION

With the rising interest in multivariable control theory brought on by increasingly complex systems the need has arisen to develop design methods that will allow the designer to specify system performance while at the same time ensuring relatively high stability margins or robustness. In the single-input single-output (SISO) case the designer has had the tools to do these tradeoffs in the form of Nyquist, Bode and root locus plots. In the multi-input multi-output (MIMO) case the classical methods are not totally appropriate.

With the increased interest in MIMO systems numerous methods of design have been employed to obtain suitable system performance and robustness with varying degrees of success. One primary method of design is to keep the plant as decoupled as possible throughout the design so that each individual element may be controlled independently and designed essentially as a single loop system. Rosenbrock [Ref. 1] has developed a procedure where the multiloop system is modified into a system that has diagonal elements that are much larger than any off-diagonal elements. This diagonally dominant system is then in a form where conventional Nyquist type techniques can be employed in the analysis. A third common MIMO design method is that of the Linear Quadratic (LQ) method. This method uses a quadratic cost functional and optimization principles to allow the designer to design for various performance levels by adjusting the matrix weighting terms used in the cost function. The major difficulty with all of the above methods is that they are not necessarily robust. This is especially true for cross-coupling terms between loops.

The primary achievement of this thesis has been the incorporation of the time domain pole placement design procedure with a method of using return difference matrix singular values to improve the robustness of the design. The technique, which utilizes a modern optimization routine, can significantly assist the designer in obtaining robustness in the face of cross-coupling perturbations. It has also been shown that the cross-coupling perturbation problem can be detected by using classical open-loop Bode diagrams as well as modern control analysis. The pole placement and robustness design routine developed for this thesis has been used on several problems discussed in recent literature. In these studies the pole placement and robustness design code has proven capable of meeting the desired goals of pole placement and robustness and also brought to light some interesting aspects of the cross-coupling perturbation problem. A slightly modified pole placement and robustness routine has proven effective in the design of robust observers.

The remainder of the thesis will present background material on SISO systems in Chapter Two and on MIMO systems in Chapter Three. Optimization will be discussed in Chapter Four along with a discussion of the Automated Design Synthesis (ADS) program used as the optimizer routine for the pole placement and robustness design procedure developed in this thesis. The thesis methodology will be discussed and outlined in Chapter Five. This will be followed by chapters discussing results from selected problems. Conclusions will be presented in the final chapter.

II. SINGLE-INPUT SINGLE-OUTPUT SYSTEMS

The purpose of this thesis research has been to develop a method of obtaining a robust multivariable control system design. A brief review of the concept of robustness and stability in the framework of a conventional SISO system will be done before pursuing the concepts in a more complicated fashion in the following chapters. A simple interpretation of robustness is the ability of the system to tolerate design perturbations. These perturbations could be in the form of actuator failures, plant parameter uncertainty, unmodeled dynamics or nonlinear terms, or any one of many other perturbations to the nominal design of the system.

The primary reason for feedback systems is the control of uncertainty within the system. By appropriate use of feedback, properties that would lead to an unstable system may be controlled. When stability and robustness aspects are considered for a SISO system, frequency domain design concepts, using either Nyquist or Bode plots, are normally used. Robustness in SISO systems is formulated naturally by the concept of gain and phase margins, both of which are readily available on the Nyquist or Bode diagram.

In figure 2.1 a nominal feedback system can be seen with a perturbation element $\alpha(s)$ placed in series with the nominal system. When $\alpha=1$ the system is nominal and stable. To determine the positive phase margin of the system the value of $\alpha(s)=\alpha(j\omega) = e^{j\varphi}$ will be changed by varying φ until the system just becomes unstable. This value of φ will then be the system phase margin. The negative phase margin can be computed in the same way. To find the gain margin the magnitude value of α is increased until the system just

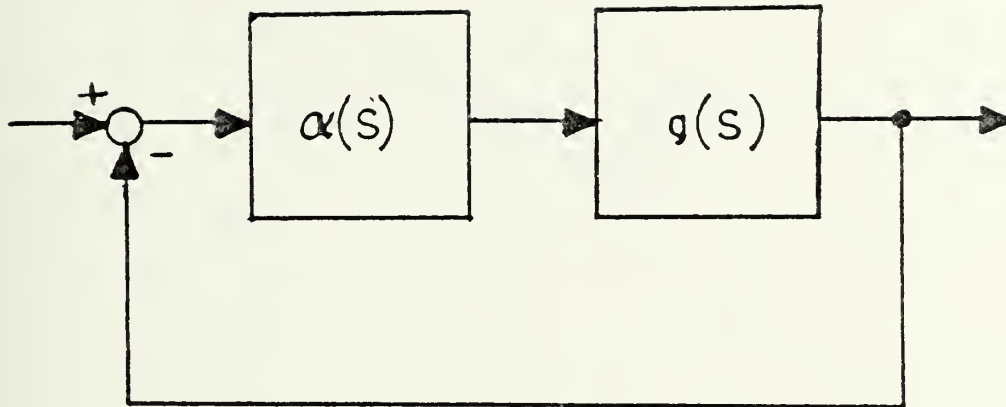


Figure 2.1 Basic Gain and Phase Margin Determination Model.

becomes unstable. This is the upward gain margin. A similar lower margin is also defined.

These gain and phase margins may not be adequate measures of robustness [Ref. 2] because they do not account for simultaneous variation in both gain and phase. Therefore, while large individual gain or phase changes may not destabilize the system, small simultaneous changes in gain and phase may destabilize the system. This is not a major difficulty in classical SISO techniques because the effect can be easily detected.

Gain and phase margin can be defined in terms of the open-loop frequency domain plots in either the Bode or Nyquist format. Figure 2.2 depicts a classical Bode plot showing gain and phase margin determination from the plot. The Nyquist plot may also be used to obtain this information. Nyquist criterion states that if the open-loop transfer function $G(s)H(s)$ has n poles in the right half plane and the limit of $G(s)H(s) = \text{constant}$ as $s \rightarrow \infty$ then for a

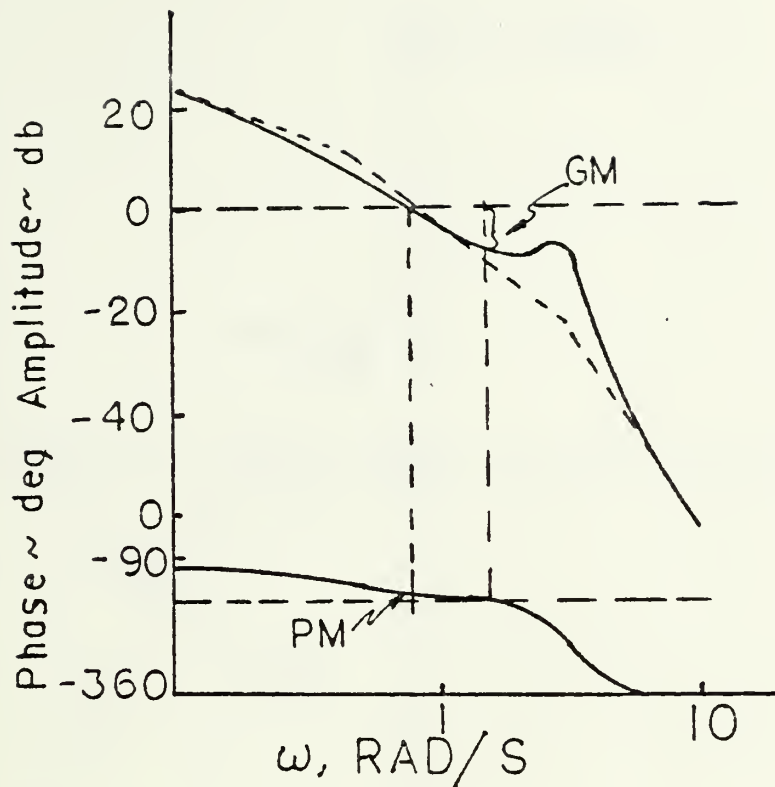


Figure 2.2 Classical Bode Plot.

stable system the locus of $G(s)H(s)$ will encircle the $-1+j0$ point n times in the counterclockwise direction as s varies along the Nyquist contour. If there are no poles in the right half s plane then the locus will not encircle the $-1+j0$ point. The diagram in figure 2.3 illustrates a nominally stable system. The gain and phase margin may be determined directly from the diagram.

Any change in the loop transfer function, provided the order of $G(s)H(s)$ does not change, that changes the number of times the locus of $G(s)H(s)$ encircles the $(-1,0)$ point in the Nyquist plot causes the system to become unstable. This leads to the conclusion that the minimum distance of the

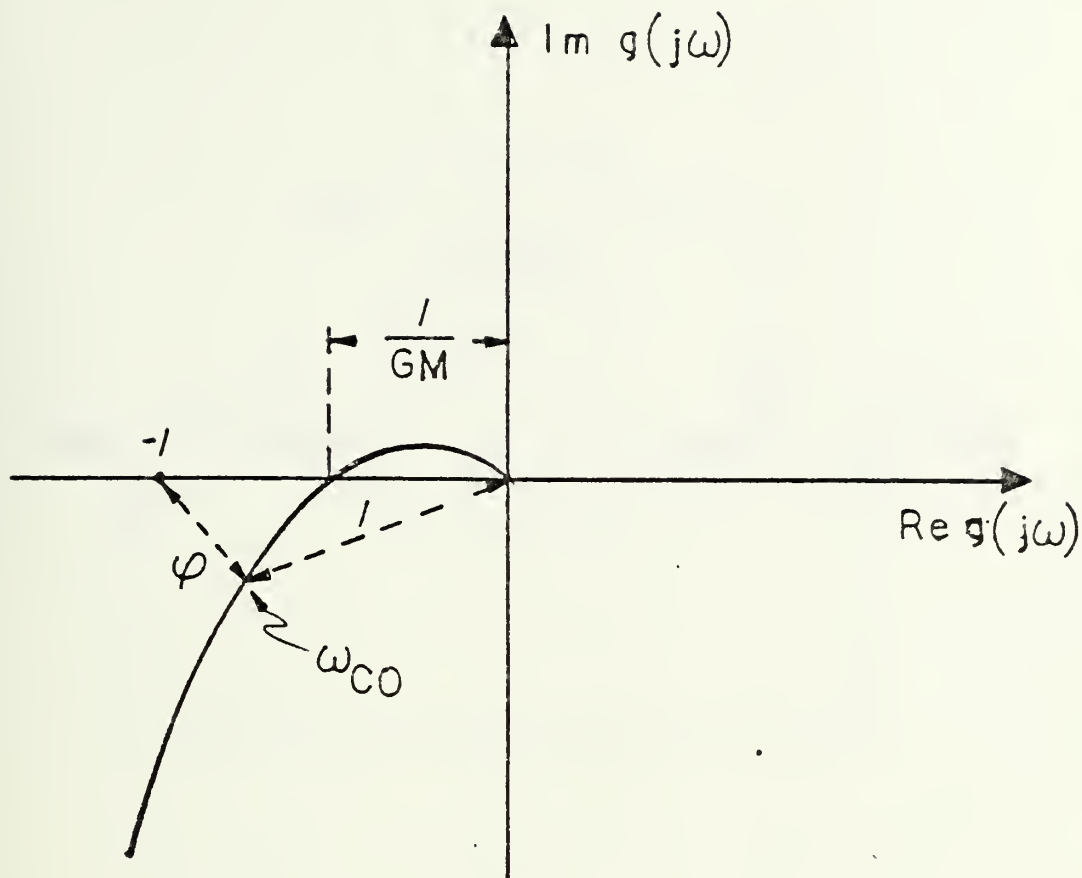


Figure 2.3 Nyquist Plot of Stable System.

locus of $G(s)H(s)$ to the $(-1,0)$ point is a measure of the system stability. This distance concept carries over directly to the MIMC system as will be shown in the next chapter. Examples of a multiplicative perturbation and an additive perturbation illustrate this idea. Figure 2.4 is an additively perturbed system. Figure 2.5 shows the Nyquist plot for this system. Assuming that the plant is itself stable and the perturbations are also stable the diagram may then be used to determine how near the system is to instability for the given perturbation.

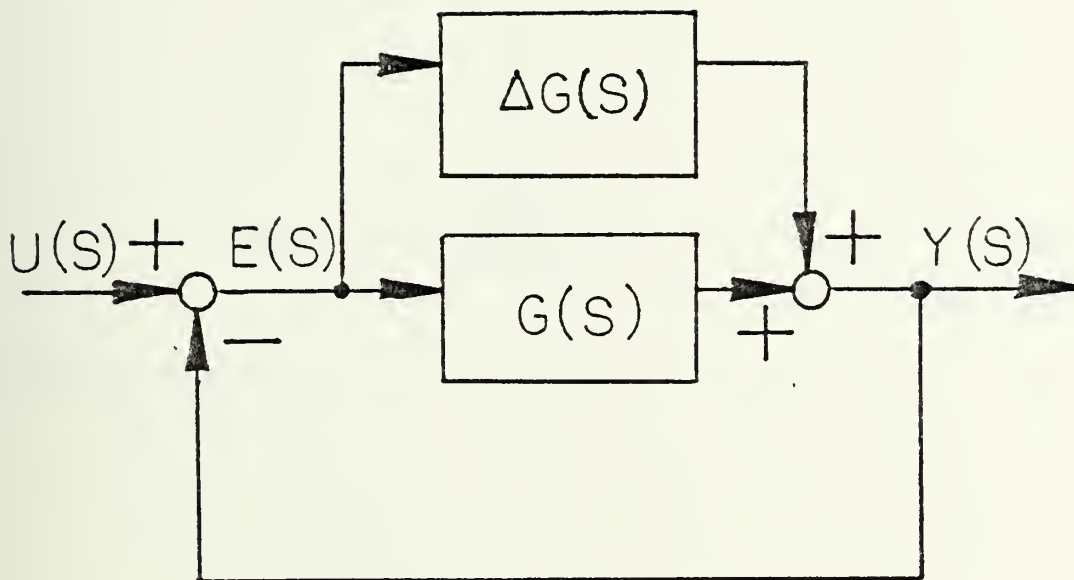


Figure 2.4 Additively Perturbed System.

Since the system is stable the $(-1,0)$ point is encircled the correct number of times by the nominal plant. If the locus of $g(j\omega)$ in the diagram is warped until it passes beyond the $(-1,0)$ point then clearly the number of encirclements of this point will change and the system will become unstable, assuming the order of the plant is not changed by the perturbation. To keep the locus of points from moving beyond the $(-1,0)$ point equation 2.1 must hold.

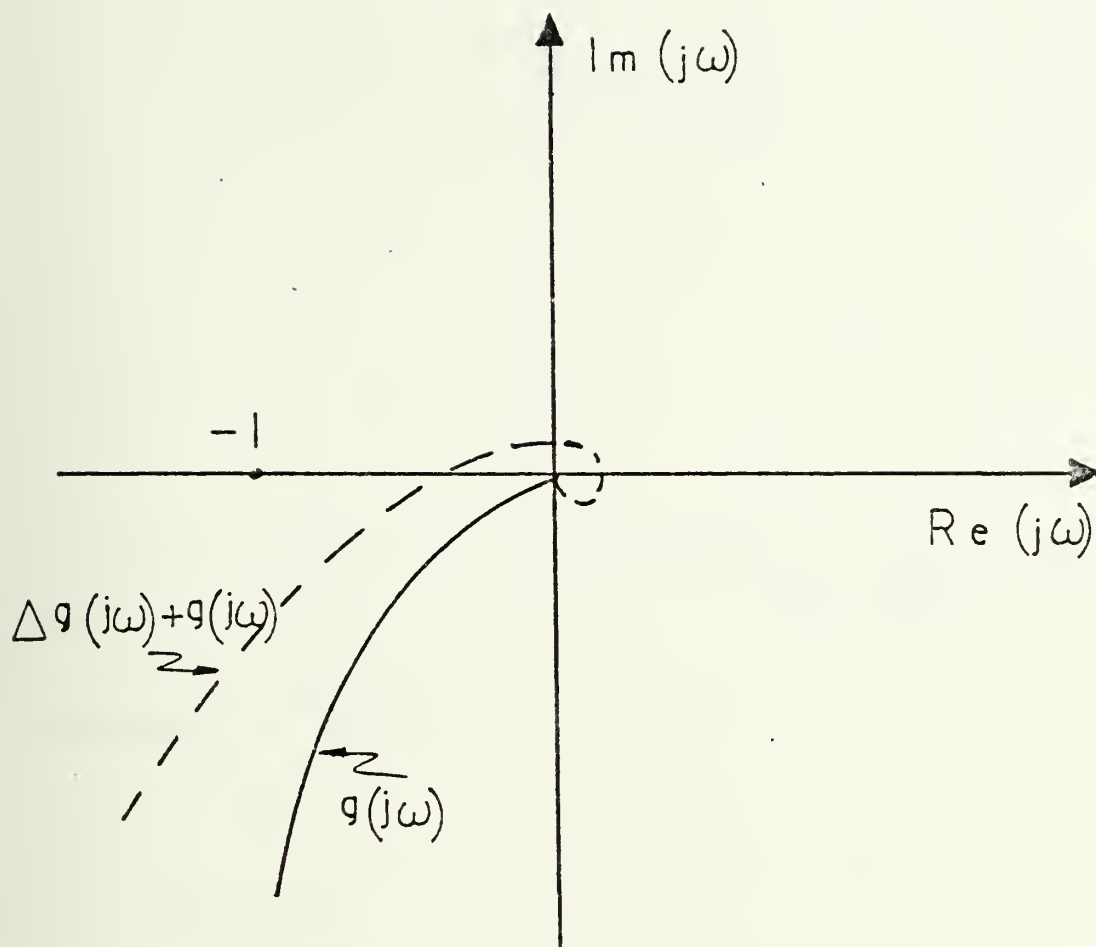


Figure 2.5 Additive Nyquist Plot.

$$|\Delta g(j\omega)| < |1 + g(j\omega)| \quad (2.1)$$

This condition is illustrated in figure 2.6. The right-hand side of equation 2.1 is just the magnitude of the return difference transfer function of the nominal system. The multiplicative case is depicted in figure 2.7 with its associated Nyquist plot in figure 2.8. The requirement for stability is similar to the additive case and may be stated in equation 2.2

$$|\Delta g(j\omega)| < |1 + (g(j\omega))^{-1}| \quad (2.2)$$

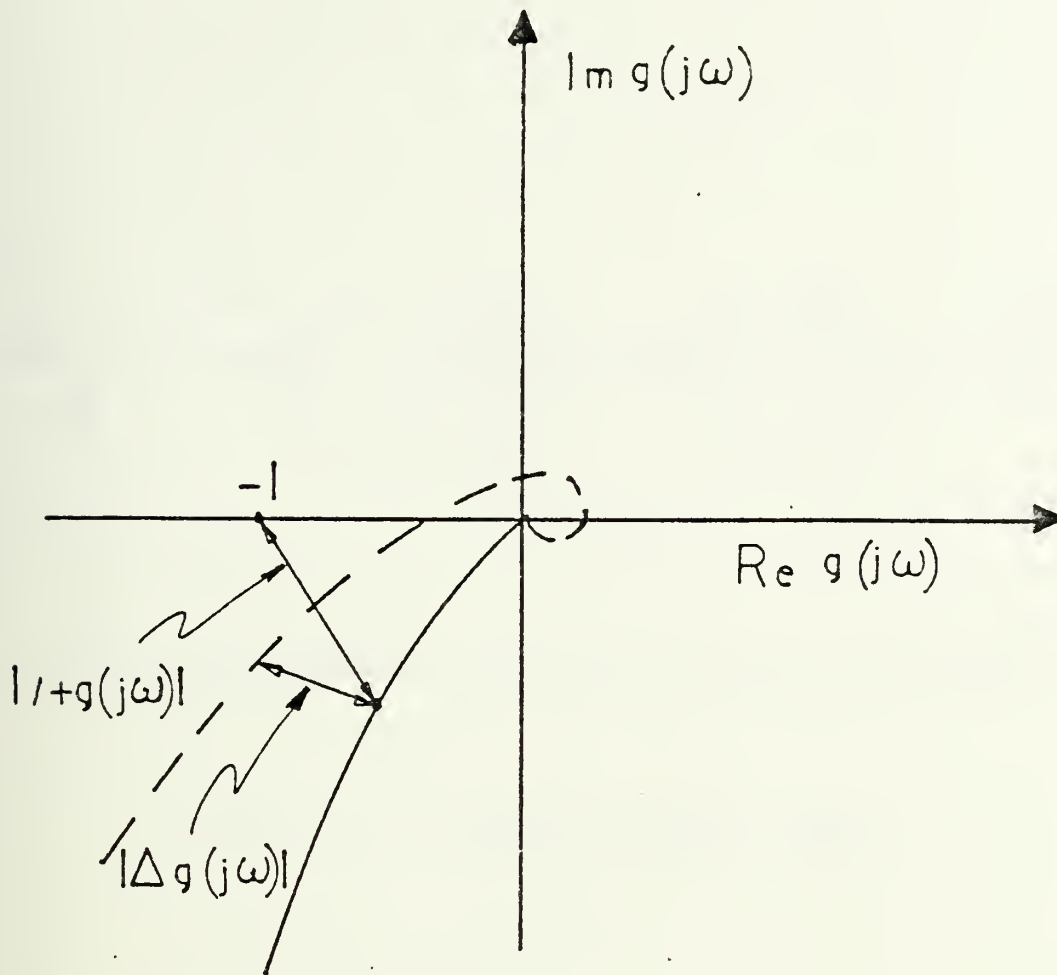


Figure 2.6 Nyquist for Inequality Additive Condition.

The above arguments will be applied again in Chapter 3 to develop multivariable stability and robustness properties.

The linear quadratic design has been the primary method employed in modern control design practice. In this method an optimal state feedback control law is developed to find a set of feedback gains that optimizes a chosen performance index. The performance index for the steady-state case is given in equation 2.3

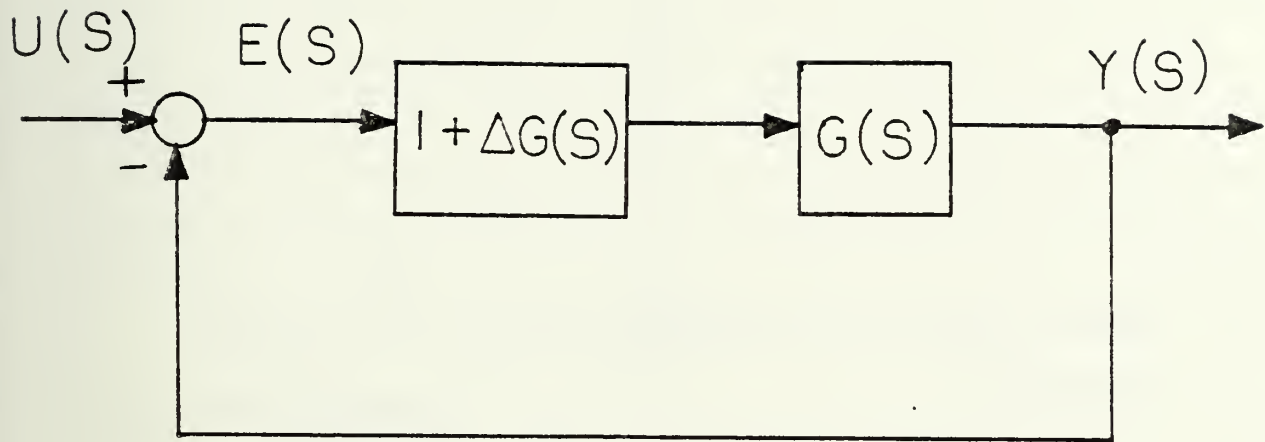


Figure 2.7 Multiplicative System.

$$PI = \int (\underline{x}' \underline{Q} \underline{x} + \underline{u}' \underline{R} \underline{u}) dt \quad (2.3)$$

where the $\underline{x}'\underline{x}$ term and the $\underline{u}'\underline{u}$ term form quadratics. The \underline{Q} and \underline{R} matrices are chosen by the designer to provide the best compromise between the minimum error of the system and the minimum energy needed to control the system. The LQ method is based on the use of closed-loop state variable feedback for the control of the system. In the MIMO problem IQ methods have been used extensively because of their guaranteed stability margins with diagonal weighting matrices. For diagonal weighting matrices the LQ method yields a guaranteed phase margin of 60 degrees and -6 db to infinite gain margin.

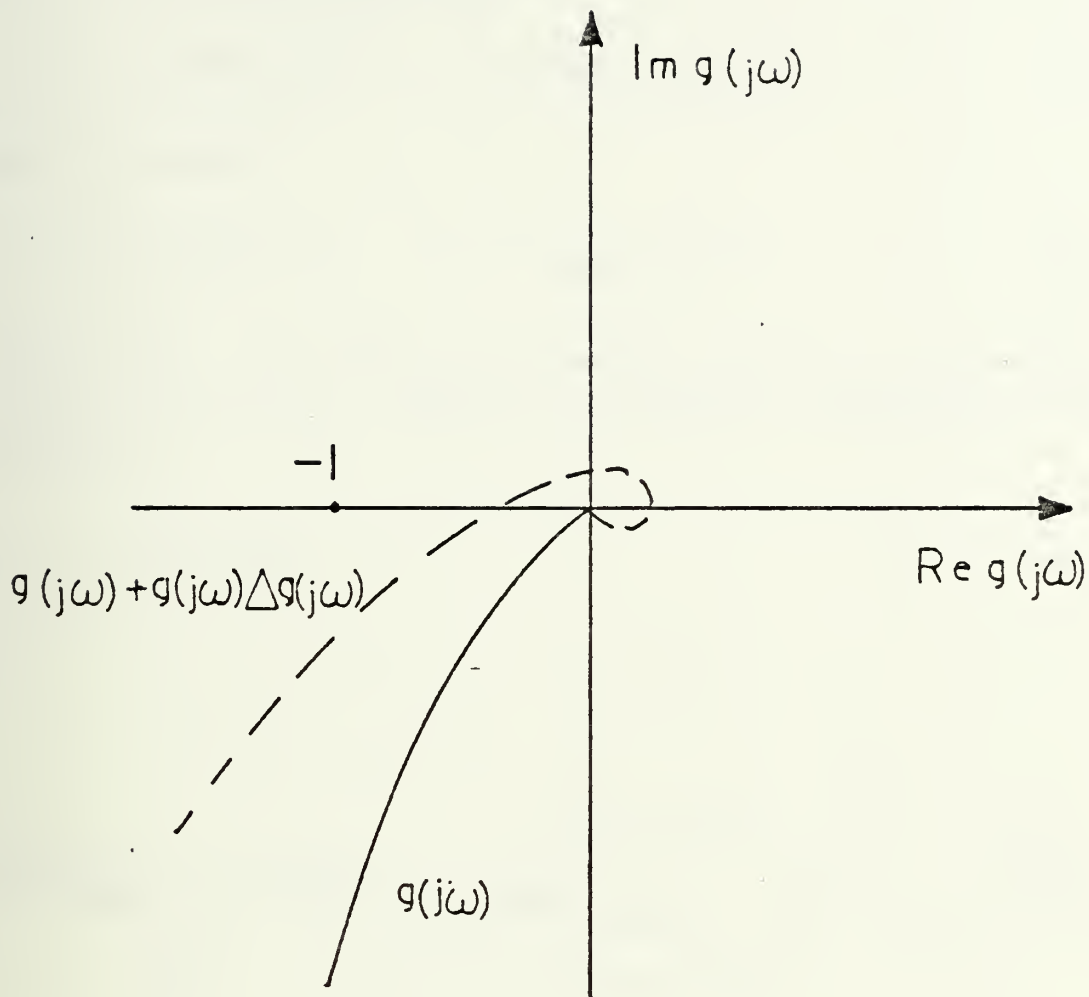


Figure 2.8 Nyquist Plot for Multiplicative System.

For SISO linear quadratic optimal regulators these stability margins can be developed from inequality 2.4 [Ref. 3]

$$|1 + f'(j\omega I - a)^{-1}b| \geq 1 \quad (2.4)$$

Writing the Kalman equation as 2.5

$$|1 + f'(sI-a)^{-1}b|^2 = 1 + (1/\rho) |G_K(s)|^2 \quad (2.5)$$

one has further

$$|1 + G(s)H(s)|^2 = 1 + (1/\rho) |G_K(s)|^2 \quad (2.6)$$

and for all $s=j\omega$ and $0 \leq \omega < \infty$ the function $(1/\rho)G_K(s)$ is greater than zero, therefore the Kalman inequality is shown to be 2.7.

$$|1 + G(s)H(s)| > 1 \quad (2.7)$$

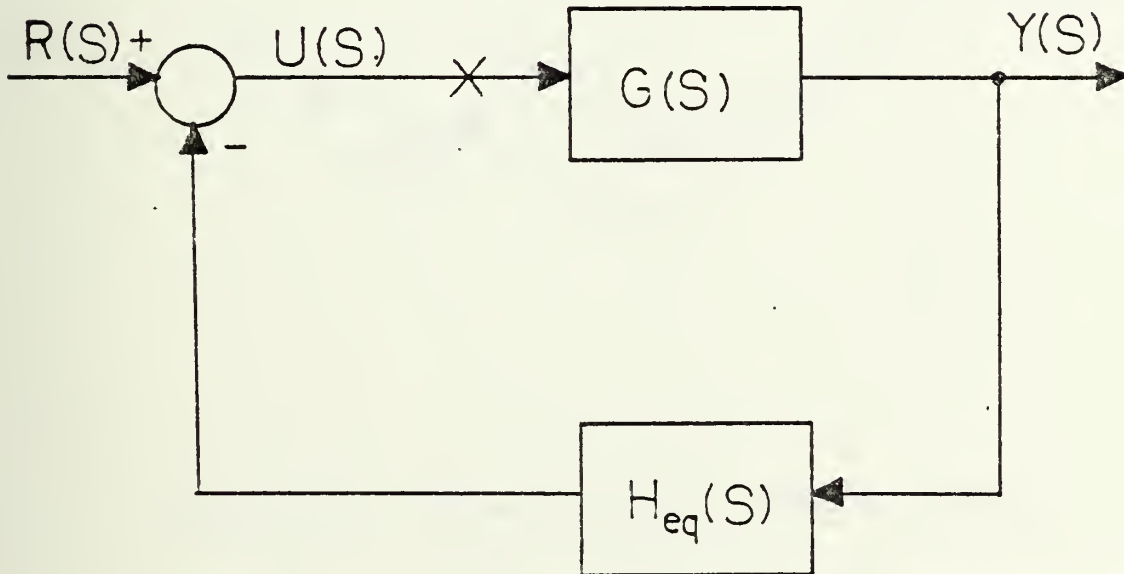


Figure 2.9 Typical Equivalent Feedback System.

The graphical interpretation of this result is that the polar plot of $G(s)H(s)$ must remain outside the unit circle centered at the $-1 + j0$ point for all frequencies. Figure

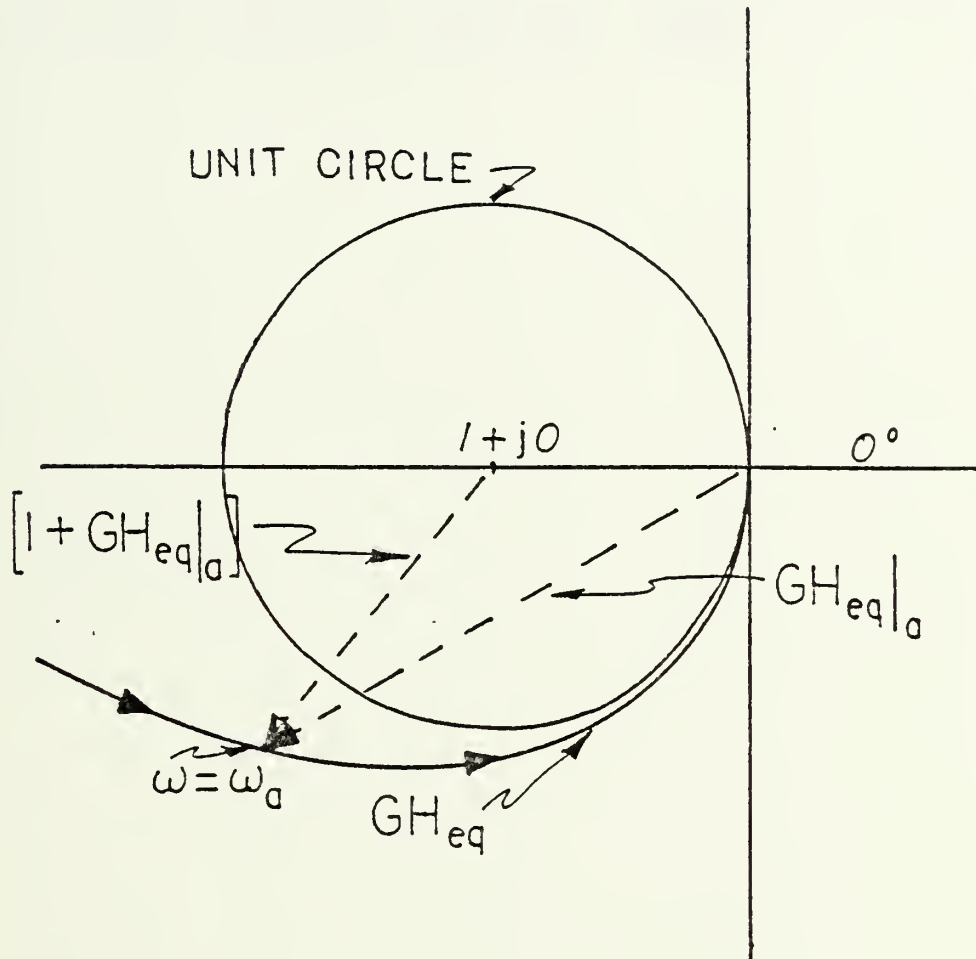


Figure 2.10 Polar Plot of an Optimal System.

2.10 shows such a polar plot. Since the optimal regulator with the loop broken at the input to the plant (denoted by the x in figure 2.9) does not penetrate the unit disk about $-1 + j0$ this means that the single input regulator will have

a phase margin of at least 60 degrees and a gain margin tolerance of fifty percent gain reduction and infinite upward margin. Further discussion of this property may be found in Anderson and Moore [Ref. 4]. With this basic review of the concepts of stability and robustness in the classical SISO system complete, the next chapter will extend some of these basic concepts to the MIMO system.

III. MULTIVARIABLE SYSTEMS

Linear quadratic design has performed relatively well in aircraft control concepts because of the ability to formulate the system state equation and to quantify acceptable performance indices for the system. Industrial applications for LQ theory have been less successful and have led British researchers to look into forms of decoupled design methods. One of the primary methods used in multivariable design is to make the system totally decoupled. This method allows each loop to be designed as a separate entity by classical means. One of the primary difficulties with this method is the problem of finding a compensator which will totally decouple the system. The method also suffers from the effects of cross-coupled perturbation terms. A method that does not totally decouple the system but makes the design problem simpler by designing a compensator that only causes the diagonal terms of the transfer matrix to be dominant over all off-diagonal terms has also been developed, [Ref. 5]. Classical frequency domain techniques are then used to design each loop of the system. The major difficulty with this and other single loop design techniques is their failure to account for cross-coupling perturbation terms that may interact between the loops. The individual loops may be highly robust in these designs but the overall system robustness may be low because of the loop interaction not accounted for in the design. This is the precise area that the singular value analysis has proven so beneficial in design. Singular value concepts may be applied to conventionally designed systems to assess their robustness. For instance, if a system is designed by LQ methods the designer may then formulate the transfer function of the system and

assess the singular values of the return difference matrix of the system. If the minimum singular value is found to be low at some critical frequencies the designer can then modify the \underline{Q} and/or \underline{R} matrices chosen in the LQ performance index and recalculate the design. In this iterative fashion a robust design would be developed.

A generalization of the SISO Nyquist theory discussed in the previous chapter has been made for the MIMO problem. This generalization leads directly to the singular value concept. The generalization is expressed in the form of the multivariable Nyquist theorem which requires that a closed loop stable system have the same number of counterclockwise encirclements of the origin by the locus of the $\det(\underline{I} + \underline{G}(j\omega))$ as the number of open loop poles that are unstable. This theorem is formally stated as;

Let $N[f(s)]$ denote the number of clockwise encirclements of $(-1, 0)$ by the locus of $f(s)$ as s traverses the contour D of figure 3.1 in a clockwise sense. The closed-loop system will be stable if and only if for all R sufficiently large

$$N[f(s)] = -P$$

where

$$f(s) = -1 + \det[\underline{I} + \underline{G}(s)] = \varphi_{oL}(s) / \varphi_{cL}(s) - 1 \text{ and}$$

$P =$ the number of closed right-half plane zeros of $\varphi_{oL}(s)$.

The application of the Nyquist theorem comes through the fact that a multivariable system will not be robust to modelling errors if the return difference matrix, $\underline{I} + \underline{G}$, is nearly singular for some frequency. If $\underline{I} + \underline{G}$ is nearly singular a small change in \underline{G} may make $\underline{I} + \underline{G}$ exactly singular. This causes the $\det(\underline{I} + \underline{G})$ to become zero and the Nyquist encirclement count to change indicating an unstable system. It is possible for very small changes in $\underline{I} + \underline{G}$ to produce large changes in the determinant of $\underline{I} + \underline{G}$. The matrix $\underline{I} + \underline{G}$

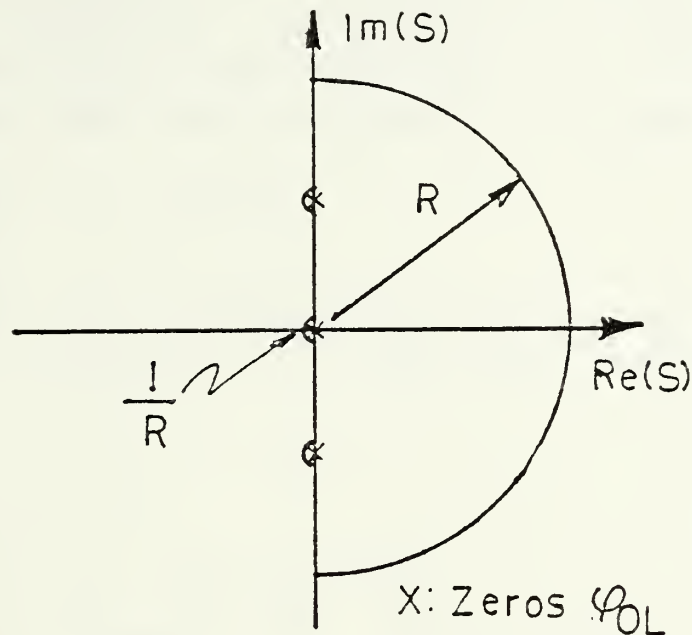


Figure 3.1 Nyquist D Contour.

$$\begin{bmatrix} 10/s+a & 9.99/s+a \\ 10/s+a & 10/s+a \end{bmatrix}$$

has determinant $0.1/(s+a)^2$. If the element p_{21} is changed by only one percent to $9.9/s+a$ the determinant becomes $1.1/(s+a)^2$ which is a significant change in the determinant value. Therefore, it is evident that $\det(\underline{I} + \underline{G})$ is not an accurate measure of how near the return difference is to singularity. Researchers, in the field of controls [Ref. 6], [Ref. 7], [Ref. 8], [Ref. 9] have used singular value analysis to determine how near the return difference matrix is to singularity.

Since the number of encirclements of the Nyquist diagram changes as $f(s)$ passes through the -1 point or when $\det(\underline{I} + \underline{G})$ is zero it is important to find how near the return difference matrix $\underline{I} + \underline{G}$ is to being singular. This nearness to singularity can be interpreted as closeness of the matrix \underline{G}

to the critical point, -1. A quantity which can be used to express the nearness to singularity of the matrix is the minimum matrix singular value denoted by $\underline{\sigma}$. Given a matrix A the singular value may be expressed by equation 3.1

$$\underline{\sigma}(A) = \min_i (\lambda_i (A^H A))^{1/2} \quad (3.1)$$

where $\lambda_i (A^H A)$ is the eigenvalue of the complex conjugate transpose of A times A . A basic MIMO linear system is

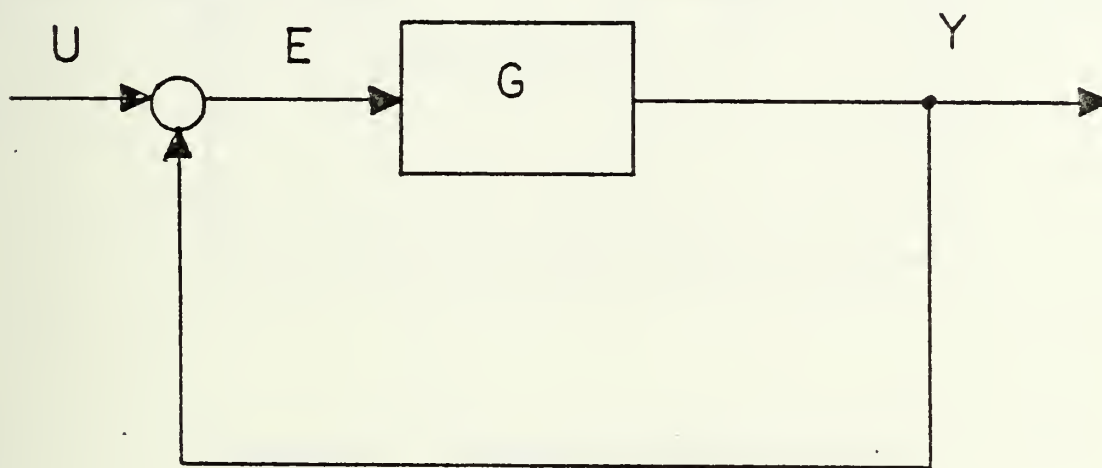


Figure 3.2 Basic Multi-input, Multi-output System.

depicted in figure 3.2. An additive perturbation to the plant is shown in figure 3.3. If the plant is stable before the perturbation is added to the system the Nyquist theorem will be satisfied and the locus of GH will not encircle the -1,0 critical point. When the perturbation is added to the system as long as the Nyquist locus is not forced to

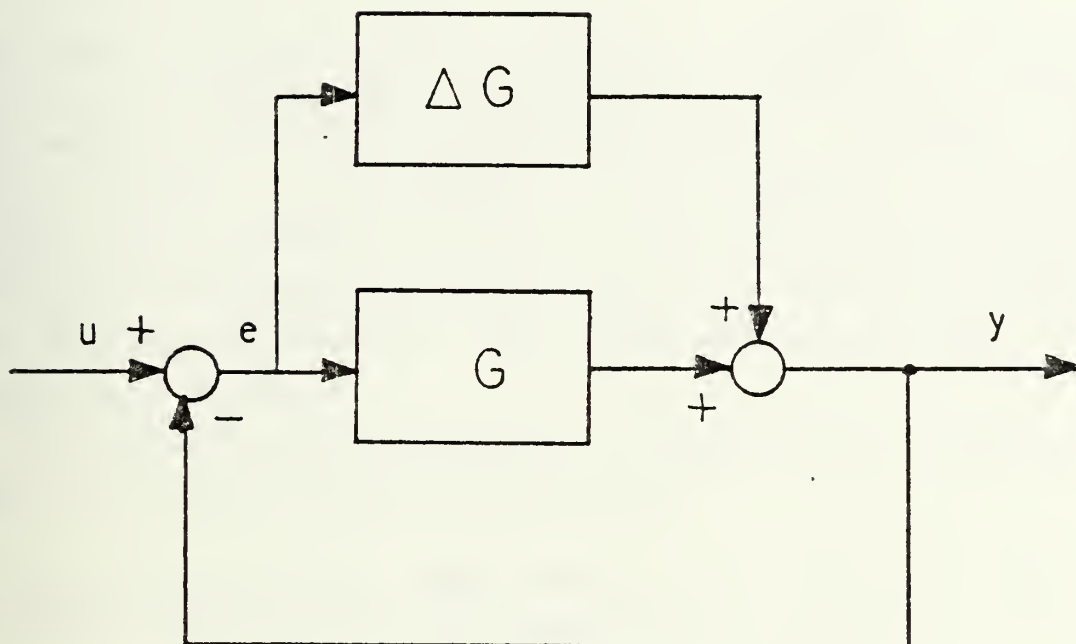


Figure 3.3 Additive Perturbation.

encircle the -1 point the system will remain stable. A sufficient condition, recalling the SISO discussion in chapter 2, for the perturbed Nyquist plot not to change encirclements is that the norm of the perturbation ΔG remain less than the norm of the return difference matrix as expressed in equation 3.2.

$$\|\Delta G(j\omega)\| < 1/\|(I+G)^{-1}\| \quad (3.2)$$

$\omega \geq 0$ This condition will guarantee that the locus of the $\det(I+G)$ does not pass through the -1 point. If the l_2 or Euclidean norm is assumed for this condition the equation

3.2 may be expressed in terms of singular values as equation 3.3.

$$\bar{\sigma}(\Delta G) \leq \underline{\sigma}(I + G) \quad (3.3)$$

This result states that as long as the maximum singular value of the perturbation matrix ΔG is below the minimum norm value of the return difference matrix the system will remain stable. The problem of guaranteeing robustness becomes that of finding the largest norm of the perturbation quantity, the largest singular value, for which the smallest norm or singular value of the return difference matrix will remain non singular.

The multiplicative form for a system such as figure 3.4 gives the similar norm equation in 3.4

$$\|\Delta G(j\omega)\| < 1 / \|(I + G)^{-1}\| \quad (3.4)$$

$\omega \geq 0$. which may be expressed as;

$$\bar{\sigma}(\Delta G) \leq \underline{\sigma}(I + G^{-1}) \quad (3.5)$$

Singular value decomposition software is readily available to determine how near the matrix $I+G$ or $I+ (G)^{-1}$ is to singularity.

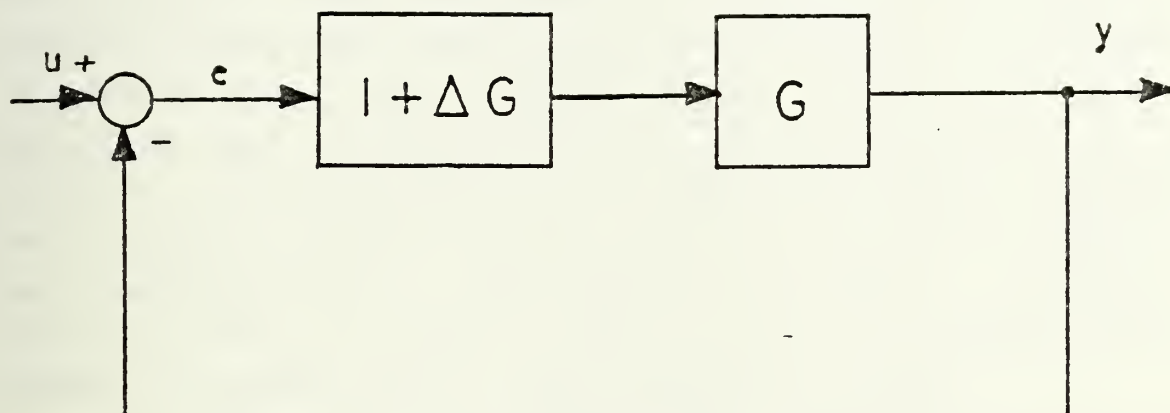


Figure 3.4 Multiplicative Perturbation.

IV. OPTIMIZATION

The purpose of this chapter will be to briefly describe several of the currently employed optimization techniques and the Automated Design Synthesis (ADS) program, [Ref. 10], which employs these techniques. In general, optimization implies finding the "best" possible solution to a problem. In actuality the best solution found by an optimization technique might really only be a "better" solution to the problem. The purpose of ADS and other optimization routines is to allow a rational search to be conducted to find the best possible design. The techniques of numerical optimization are used to logically vary the various parameters that affect the design until a good solution is found.

As an example of an unconstrained optimization problem consider the following problem developed in [Ref. 11]. The problem is to minimize the function

$$F(x) = 10x_1^4 - 20x_1^2x_2 + 10x_2^2 + x_1^2 - 2x_1 + 5 \quad (4.1)$$

$F(x)$ is often called the objective function, the cost function or the penalty function. Since there are no conditions imposed on the design variables, x_1 , and x_2 , and no additional limits imposed on the overall design, the problem is considered to be one of unconstrained minimization. Figure 4.1 represents this problem in the design space. From the figure it appears the optimum is near the point 1,1. Calculus may be applied to determine the optimum exactly. Taking the derivatives

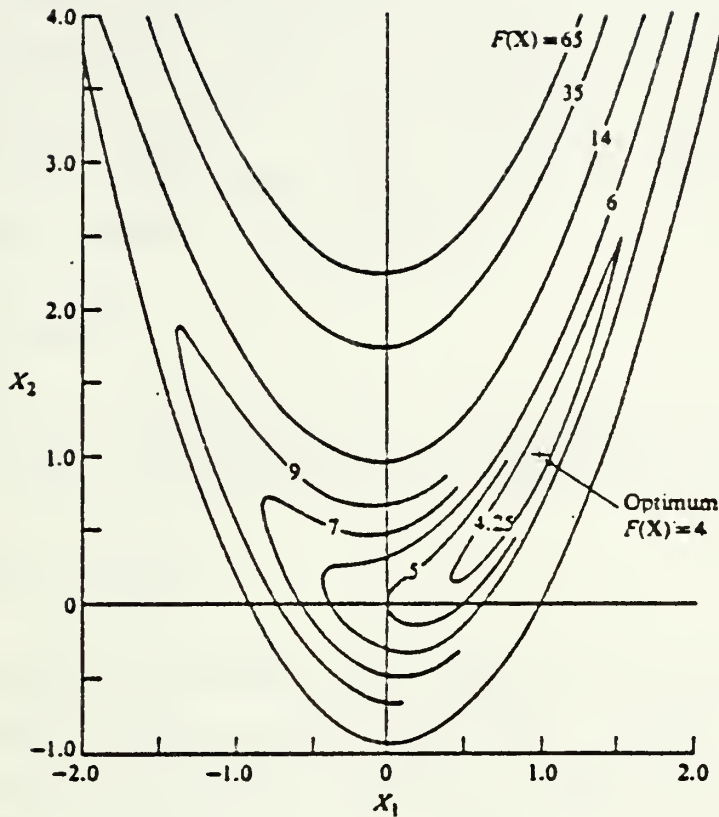


Figure 4.1 Design Space for Example Problem.

$$\partial F(\underline{x}) / \partial x_1 = 40x_1^3 - 40x_1x_2 + 2x_1 - 2 = 0 \quad (4.2)$$

$$\partial F(\underline{x}) / \partial x_2 = -20x_1^2 + 20x_2 = 0 \quad (4.3)$$

and then solving the set of equations it is found that $x_1 = 1.0$ and $x_2 = 1.0$.

If design conditions are imposed on the problem then the optimization becomes one of constrained function minimization. In other words, while the minimum of a function is still sought, this minimum must exist within the limits

imposed by the design conditions or constraints. The minimum of an unconstrained function will not necessarily be the same minimum for a constrained function.

As another example , also from [Ref. 11], a design is sought which gives the minimum weight of a particular column. This weight is expressed as:

$$W = \rho A h = \rho \pi D h \quad (4.4)$$

where ρ is the unit weight of the material and A the cross sectional area. The stress in the column is given by:

$$\sigma = P/A = P/\pi D t \quad (4.5)$$

without going into detail the design will be constrained by the allowable stress on the structure. Other constraints , Euler buckling and shell buckling, are also of interest to the designer of this column. The design problem is then stated as:

$$\text{minimize } W = \rho \pi D h \quad (4.6)$$

for the constraints

$$g(1) = \sigma / \bar{\sigma} - 1 \leq 0 \quad (4.7)$$

$$g(2) = \sigma / \sigma_0 - 1 \leq 0 \quad (4.8)$$

$$g(3) = \sigma / \sigma_s - 1 \leq 0 \quad (4.9)$$

with $D \geq 10e-06$ and $t \geq 10e-06$.

$$g(4) = t - D \leq 0$$

(4.10)

$\bar{\sigma}$, σ_b , σ_s refer to the allowable design stress, Euler buckling stress and the shell buckling stress respectively.

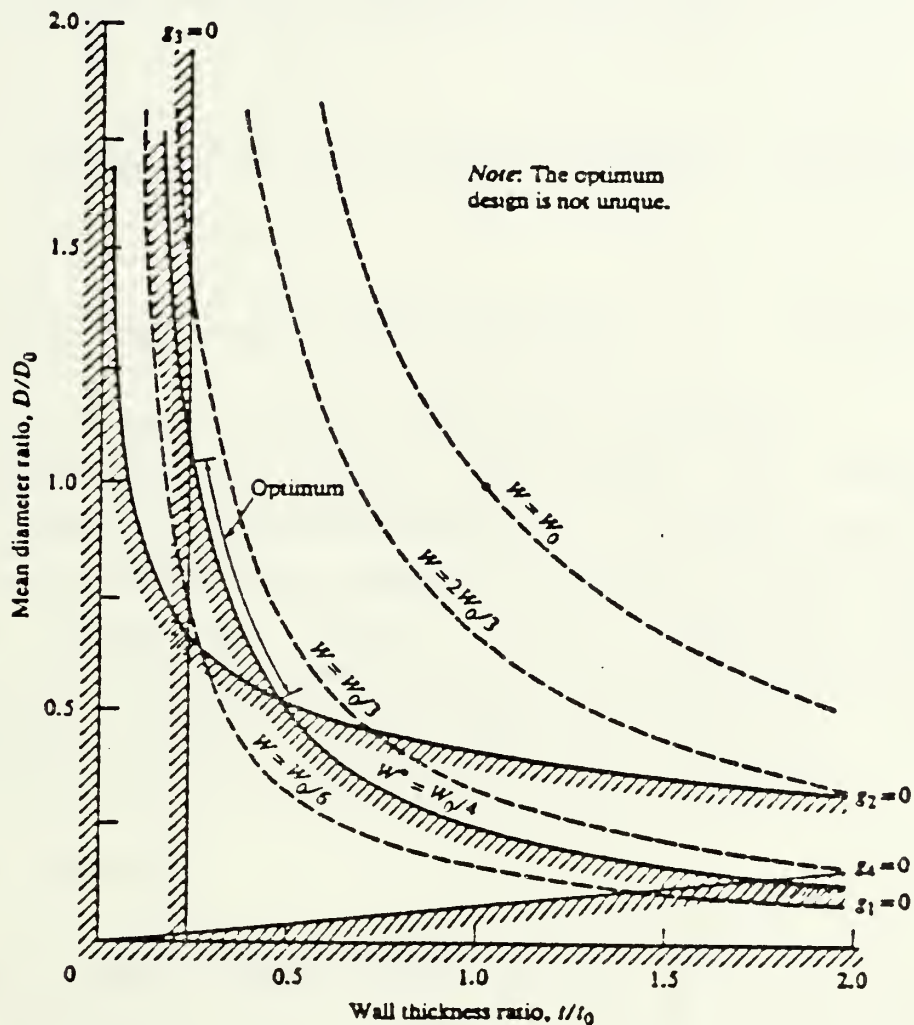


Figure 4.2 Design Space for Column Problem.

Figure 4.2 is a design space diagram for this problem. It is noted from the figure 4.2 that the optimum is not unique and can be any value along the arc noted as optimum in the figure.

The general optimization problem is then written as:

$$\text{minimize } F(\underline{x}) \quad \text{objective} \quad (4.11)$$

$$\text{subject to } g_j(\underline{x}) \leq 0 \quad j=1, m \text{ inequality constraint} \quad (4.12)$$

$$h_k(\underline{x}) = 0 \quad k=1, l \text{ equality constraint} \quad (4.13)$$

$$x_i^l \leq x_i \leq x_i^u \quad i=1, n \text{ side constraint} \quad (4.14)$$

where $\underline{x} = \text{col}(x_1, x_2, \dots, x_n)$ is the design variable.

The methods used to solve this problem are usually iterative. After the establishment of an initial set of variables the optimizer will update this initial value until the optimum values are found. Again, borrowing from [Ref. 11], the iterative technique may be demonstrated by a simple example. Figure 4.3 is used to illustrate this problem. Given the initial data set \underline{x}^0 the formula

$$\underline{x}^1 = \underline{x}^0 + \alpha^* \underline{s}^1 \quad (4.15)$$

can be used to upgrade this estimate of \underline{x} . The vector \underline{s} is the search direction for the iteration and the scalar quantity α is the distance of the move in the \underline{s} direction. Beginning at \underline{x}^0 it is desired to reduce the objective function. The search for values of \underline{x} that reduce the objective function is made in the \underline{s} direction which in this example is the opposite of the gradient of the function at point \underline{x}^0 . The choice of \underline{s} could be arbitrary as long as it reduces the

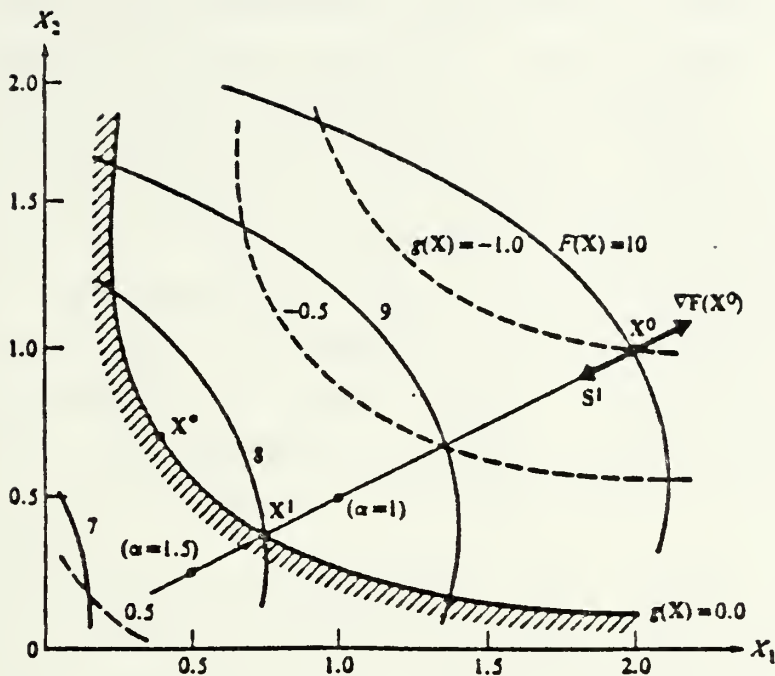


Figure 4.3 Illustrative Example for Iteration.

value of the objective. If \underline{S} is chosen to be opposite the gradient of the objective then the search would be a steepest descent search. Once the direction is chosen then the scalar must be found to minimize the objective along this direction vector without crossing a constraint boundary. The technique for finding α is to evaluate \underline{X} , the objective and the constraint functions for values of α using computer code and numerical interpolation to estimate α . This α value then gives the minimum value for $F(\underline{X})$ along this search vector. Now that α for this direction is known the new equation becomes:

$$\underline{x}^2 = \underline{x}^1 + \alpha^* \underline{S}^2 \quad (4.16)$$

and now \underline{x}^1 is used to start a new search in direction \underline{S}^2 and compute a new α^* that continues to reduce the objective

function. These one dimensional searches are continued until no more reduction in the objective can be found. At this point no further design improvement is possible.

Optimization techniques do not always lead to the absolute optimum when applied to problems of practical interest. The reasons for this could be numerical ill-conditioning of the problem formulation or simply that there are multiple solutions to the problem. Because of these difficulties it may be advisable to choose several different starting points for the optimization process and use engineering judgement as to the design most applicable to the problem under analysis.

Considering the unconstrained case first where the desire is to minimize the function $F(\underline{x})$, it is well known that $F(\underline{x})$ will have a minimum where the gradient of $F(\underline{x})$ is zero. That is:

$$\text{grad}(F(\underline{x})) = 0 \quad (4.17)$$

with the $\nabla F(\underline{x})$ defined as:

$$\text{grad}(F(\underline{x})) = (\partial F(\underline{x}) / \partial x_1, \dots, \partial F(\underline{x}) / \partial x_n)' \quad (4.18)$$

Figure 4.4 shows why this is a necessary condition but does not guarantee a global minimum. The gradient of $F(\underline{x})$ is zero at all four points A, B, C, and D. However, only A and D are minima. A would be the global minimum for the function as defined here. D would be only a relative minimum. To check that the zero gradient corresponds to a minimum the Hessian matrix, i.e. the matrix of second partial derivatives, can be examined for positive definiteness. A positive definite Hessian ensures a relative minimum. The only way to prove a global minimum for the function is to show that the Hessian matrix is positive definite for all design variables \underline{x} . A test that is seldom possible to perform.

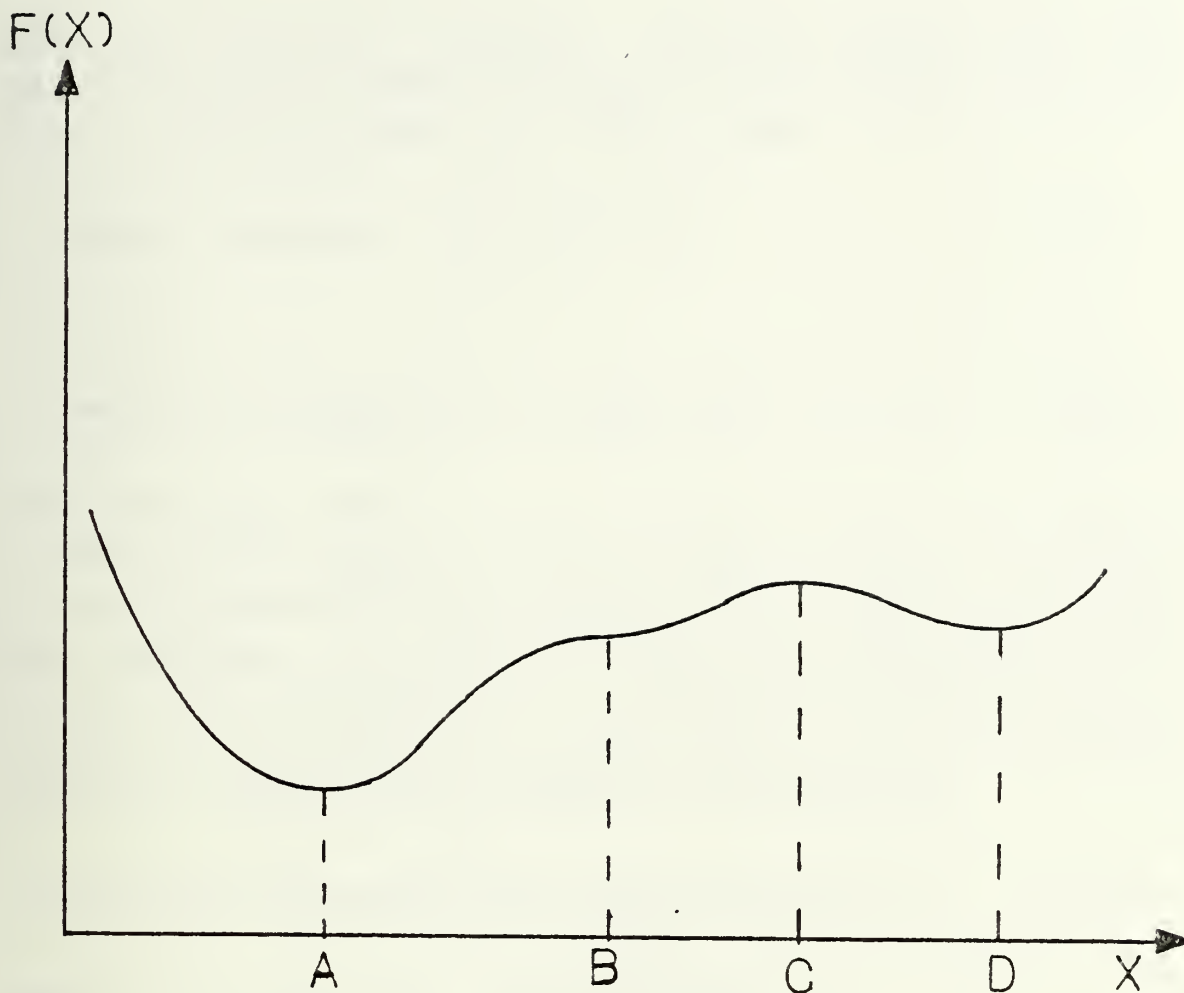


Figure 4.4 Plot of Various Points With Zero Gradient.

If the problem of minimization is a constrained problem the situation is different from that discussed above. The objective function gradient does not have to be zero at the optimum. Figure 4.5 illustrates this case. Using figure 4.5 and assuming a start at point A it is necessary to choose a search direction S which will reduce the objective function while not violating the constraint functions. Any direction that will reduce the objective is said to be useable. This is seen to be the half plane sector below the tangent to $F(x)$ at point A. If a half plane to the right of the tangent to the active constraint at point A is considered then the feasible sector is formed. A combination of the two

conditions gives the useable, feasible sector where the search direction \underline{S} must be chosen. In mathematical nomenclature the above argument may be stated as:

$$\text{useable direction } \text{grad}(F(\underline{x})) \cdot \underline{S} \leq 0 \quad (4.19)$$

$$\text{feasible direction } \text{grad}(g_j(\underline{x})) \cdot \underline{S} \leq 0 \text{ for all } j \quad (4.20)$$

for which $g_j(\underline{x}) = 0$.

Point B in figure 4.5 shows a point where the gradient of the objective and constraint point in exactly the opposite direction. At a point such as B the only search vector \underline{S} that meets requirements for useability and feasibility is tangent to the constraint boundary and to a line of constant objective function. This condition is stated as:

$$\text{grad}(F(\underline{x})) + \sum \lambda_j \text{grad}(g_j(\underline{x})) + \sum_{m+k} \lambda_{m+k} \text{grad}(h_{m+k}(\underline{x})) = 0 \quad (4.21)$$

where $\lambda_j \geq 0$ and λ_{m+k} -unrestricted.

With this brief background in optimization complete the Automated Design Synthesis (ADS) program will be briefly discussed. More detail on the ADS routine can be found in [Ref. 10]. This code was developed as a follow-on to the successful CONMIN code [Ref. 12] developed by Vanderplaats. It is designed as a black box optimizer which allows the user to choose combinations of one dimensional search, optimization algorithm and optimization strategy. These will be discussed later. For the user with specific requirements the code may be tailored by parameter modification to meet specific requirements. For the work done in this thesis it was assumed the user of the code has no detailed knowledge of optimization and will want to use the code in the

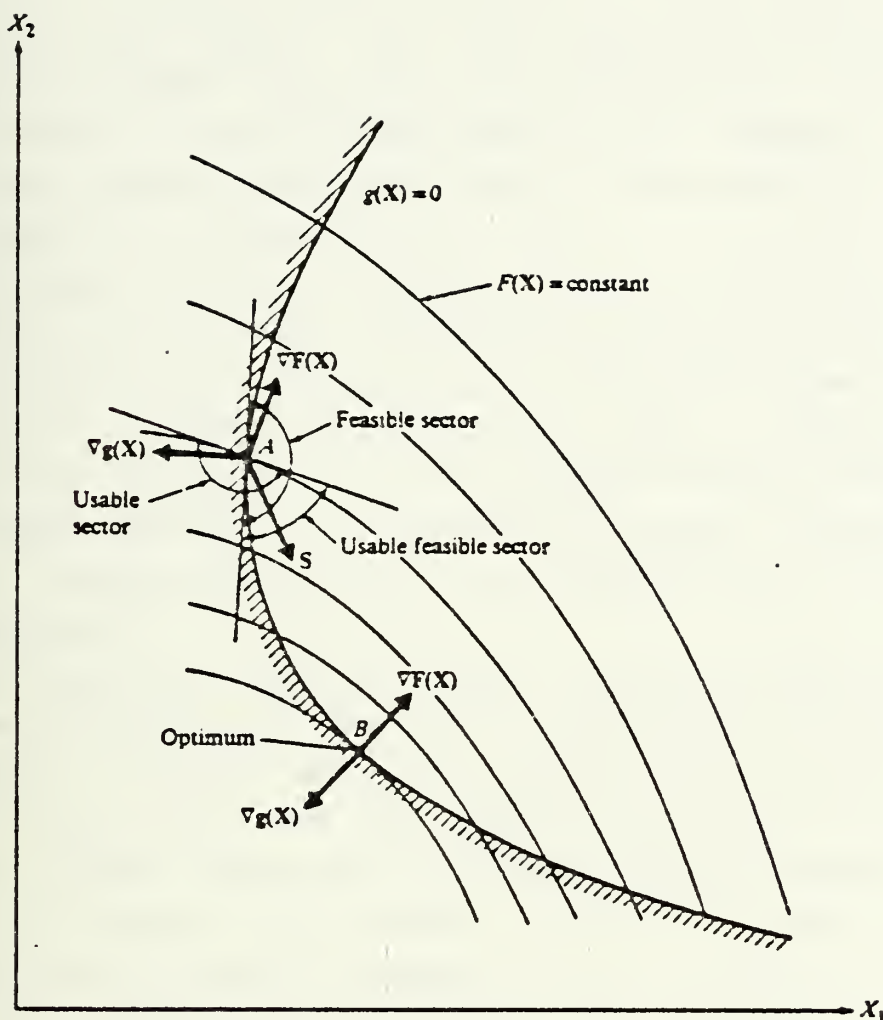


Figure 4.5 Constrained Optimization Example.

simplest form. As such, the ADS calls made from the main program use only default parameters and first forward finite difference gradients. Should analytical gradients be available the user could use them within the code if desired with no difficulty. The calls made by the user to the ADS routines specify several important aspects of the problem solution. The user may call for an optimization strategy to be used in the routine. This is not required and its use

depends on the problem. These strategies are discussed in [Ref. 10]. Two strategies used most often in the analysis done for this thesis have been sequential unconstrained minimization using quadratic exterior penalty function and the augmented Lagrange multiplier (ALM) method. Others available include sequential linear programming and sequential quadratic programming.

The basic optimizer is also chosen by the user from two unconstrained and three constrained optimization algorithms. The unconstrained algorithms are Fletcher-Reeves conjugate directions, Davidon-Fletcher-Powell (DFP) variable metric method and the Broyden-Fletcher-Goldfarb-Shanno (BFGS) variable metric method. The method of feasible directions and robust feasible directions are available for constrained minimization.

The user has available several types of one dimensional searches using Golden Section or polynomial approximation techniques. The ADS code has tailored these one dimensional search algorithms for the unconstrained and constrained cases, allowing the user to make appropriate choices for the type of problem to be solved. Figure 4.6 shows the basic organization of the ADS program.

Since it is not the purpose of this chapter to cover the subject of optimization in detail all the possible routines in ADS will not be discussed. A few of the routines found to work well for the work completed in this thesis will be briefly described, however. There are several methods of optimizing functions of one variable or one dimensional searches. For instance a large number of points, n , could be chosen and the function $F(x)$ evaluated at each point. The point corresponding to the smallest value of $F(x)$ could then be considered the optimum value of the one dimensional search. This method is hit and miss and better methods of

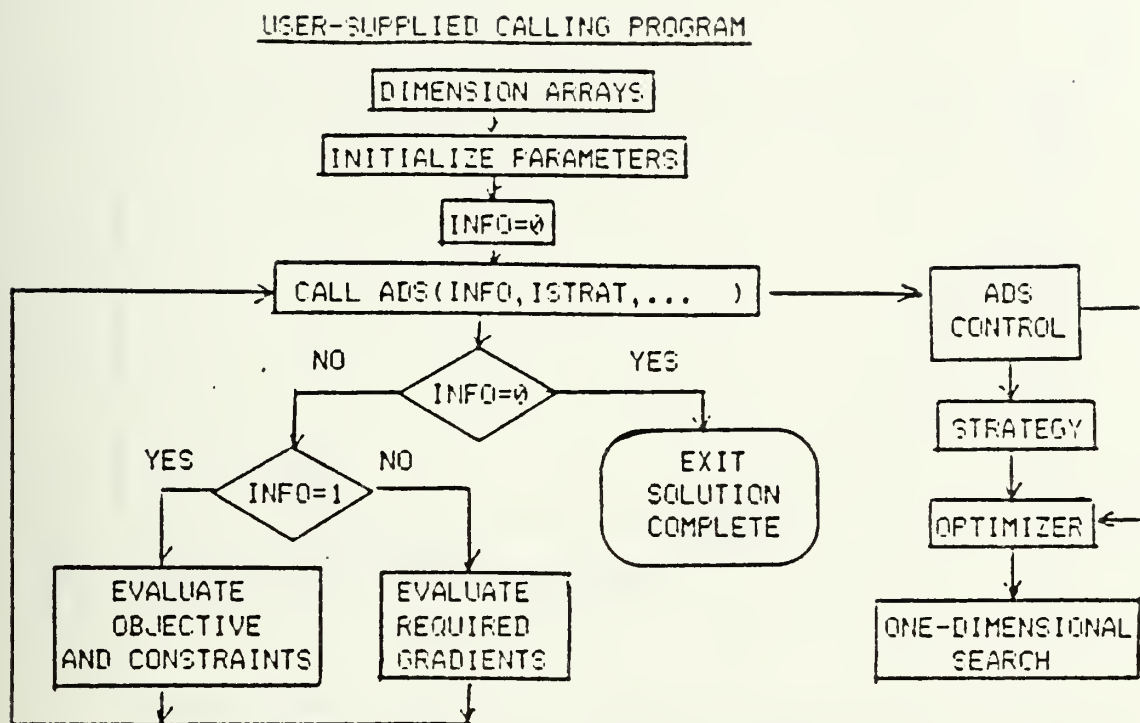


Figure 4.6 Organization of ADS Program.

locating optimum points can be used. The methods used in ADS are Golden Section and polynomial interpolation.

Golden Section search methods are easy to program on the digital computer and do not require continuous derivatives. They have a known convergence rate and are reliable for poorly conditioned problems. The major drawback of the Golden Section routines is the large number of function evaluations require. The Golden Section algorithm is simply illustrated through the use of figure 4.7 Assume that X^0 and X^1 are known to be bounds on the curve's minimum value. Also, the function values of $F(x)$, F^0 and F^1 , are evaluated and known at these points. By picking two intermediate points X^2 and X^3 where $X^2 < X^3$ and evaluating the function

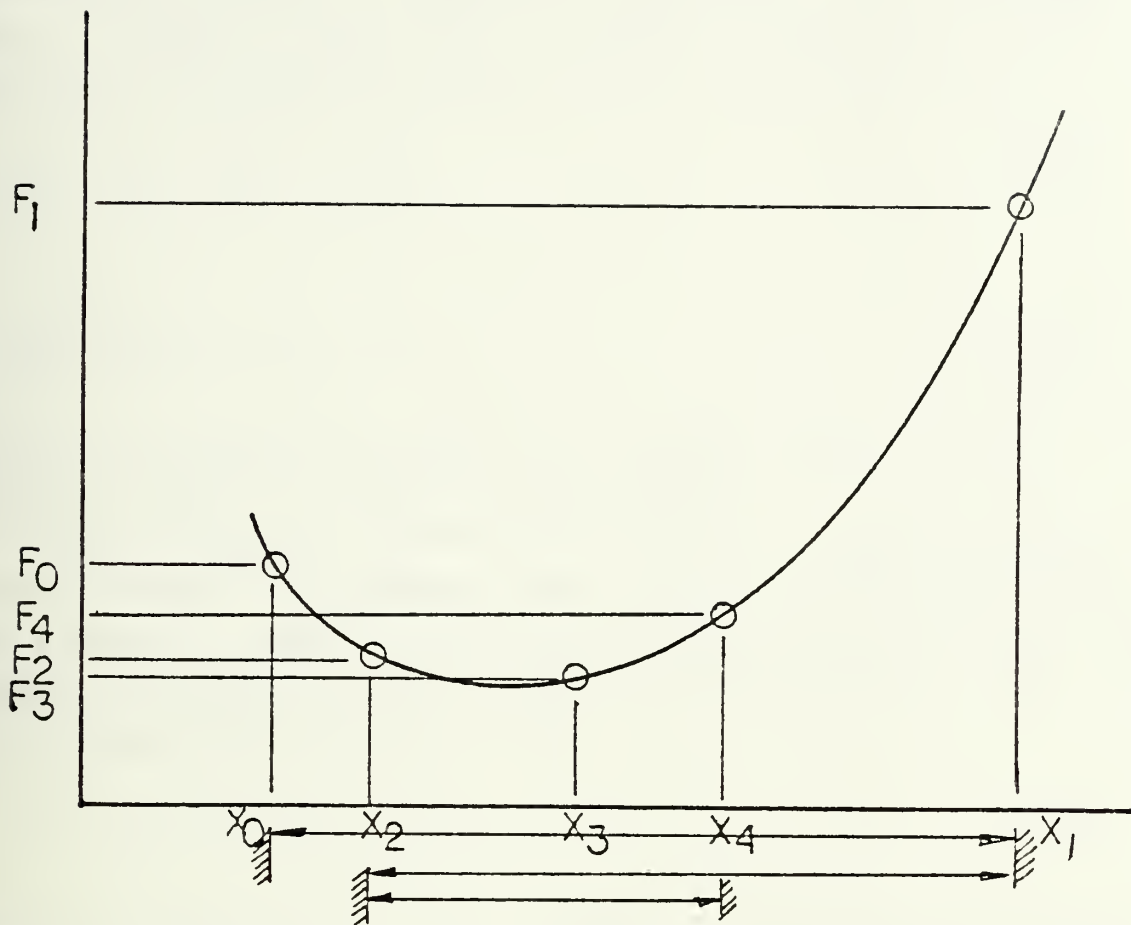


Figure 4.7 Golden Section Diagram.

at these points a bound on the minimum may be modified. Since F^2 for this figure is larger than F^3 the point X^2 forms the new lower bound. The minimum is now between X^2 and X^1 . If the function F^4 at X^4 is determined and shown to be larger than F^3 then X^4 becomes the new upper bound on the minimum. By repeating this procedure the bounds may be narrowed to any desired tolerance. The Golden Section rule is applied to this problem to reduce the bounds in the quickest possible time. By appropriately picking the values of X 's at which each function evaluation is made an efficient algorithm that uses the ratio proportion of the Golden

Section, i.e. $X^2/X^1 = 1.62803$ is developed. By choosing a value τ based on this Golden Section rule where $\tau = 0.38197$ estimates for interior points X^2 and X^3 can be made as:

$$X^2 = (1 - \tau) X^0 + \tau X^1 \quad (4.22)$$

$$X^3 = \tau X^0 + (1 - \tau) X^1 \quad (4.23)$$

As each new bound is found the process repeats until the accepted level of convergence is reached.

The polynomial interpolation method is accomplished by first fitting a polynomial curve to the points about where the minimum is desired and then finding the minimum of the polynomial function. For example if the function $F(X)$ is approximated by a quadratic as:

$$F = a_0 + a_1 X + a_2 X^2 \quad (4.24)$$

Then the value of X , X^* , where F' is zero can be shown to be :

$$X^* = -a_1 / 2a_2 \quad (4.25)$$

If a_2 is positive then F will be minimum. Other degrees of polynomials may be used in similar fashion.

Now that the basic one dimensional search methods have been reviewed the next step is to examine the basic optimization routines. First, the unconstrained case will be reviewed. The optimum X^* is at the point where

$$\text{grad}(F(\underline{X}^*)) = 0$$

(4.26)

Several zero order methods exist for the purpose of finding the minimum value. These include random search, Powell's method and Box's method. Since ADS does not use these methods they will not be discussed.

Automated Design Synthesis makes use of first order methods which will now be discussed. The steepest descent method is best known, but poor in performance. Just as the name implies a search direction is chosen opposite the gradient of the objective function. At iteration 0

$$\underline{S}^0 = -\text{grad}(F(\underline{X}^0))$$

(4.27)

Figure 4.8 shows this algorithm geometrically. Note that the method simply stair steps its way down the "hill" to the valley or minimum. ADS uses the Fletcher-Reeves modification to steepest descent. In this routine a conjugate direction is chosen to improve the speed of the search. Figure 4.9 shows how this method tracks to the solution.

The variable metric methods listed earlier are usually more powerful than Fletcher-Reeves because they store information that allows the algorithm to approximate the inverse of the Hessian matrix or second derivative. For further discussion of these methods see [Ref. 11] or other similar optimization texts.

ADS employs two direct methods for constrained minimization. One method is that of feasible direction and the other is the method of robust feasible direction. Since these two methods were seldom employed within the work presented in this paper they will not be discussed.

The methods chosen to handle the constrained minimization problems formulated in this thesis are referred to as Sequential Unconstrained Minimization Techniques (SUMT).

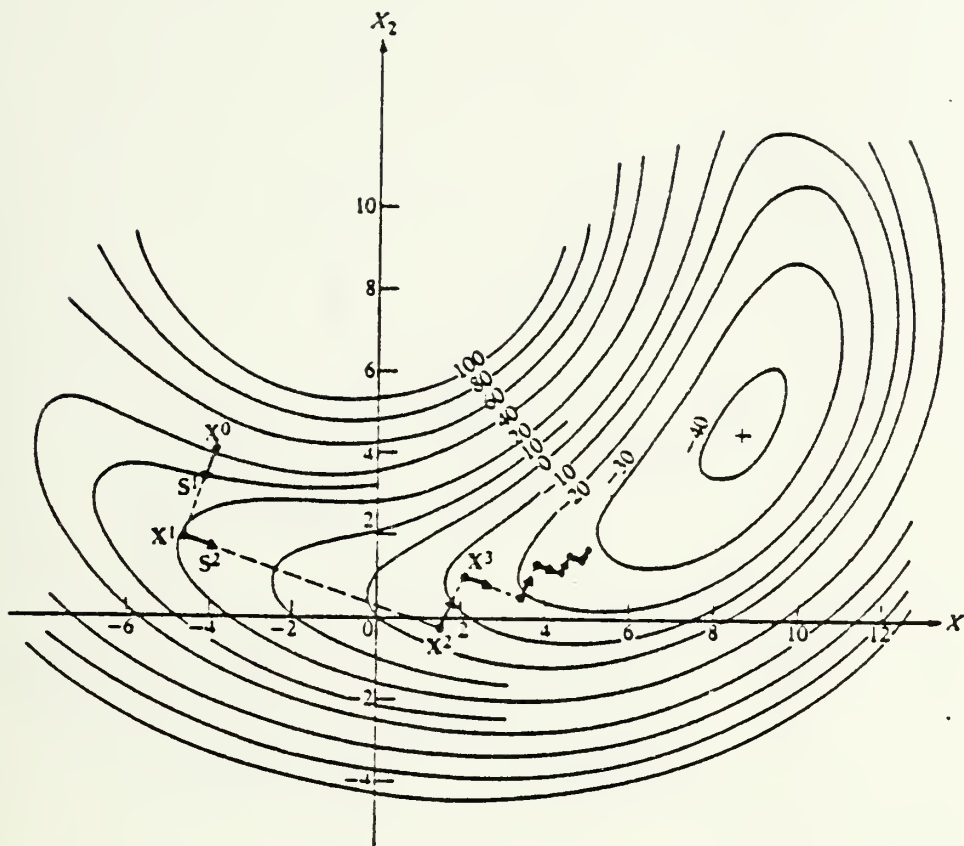


Figure 4.8 Steepest Descent Algorithm.

SUMT methods are methods which formulate the objective function and the constraint functions into an augmented objective function and then solving the problem as if it were an unconstrained optimization task. ADS employs interior and exterior penalty function techniques as well as an Augmented Lagrange Multiplier (ALM) technique.

The exterior penalty function method is incorporated by forming a penalty from the constraint equations. This penalty is of the form :

$$P(\underline{x}) = \sum_{j=1}^m \max(0, g_j(\underline{x}))^2 + \sum_{k=1}^l (h_k(\underline{x}))^2 \quad (4.28)$$

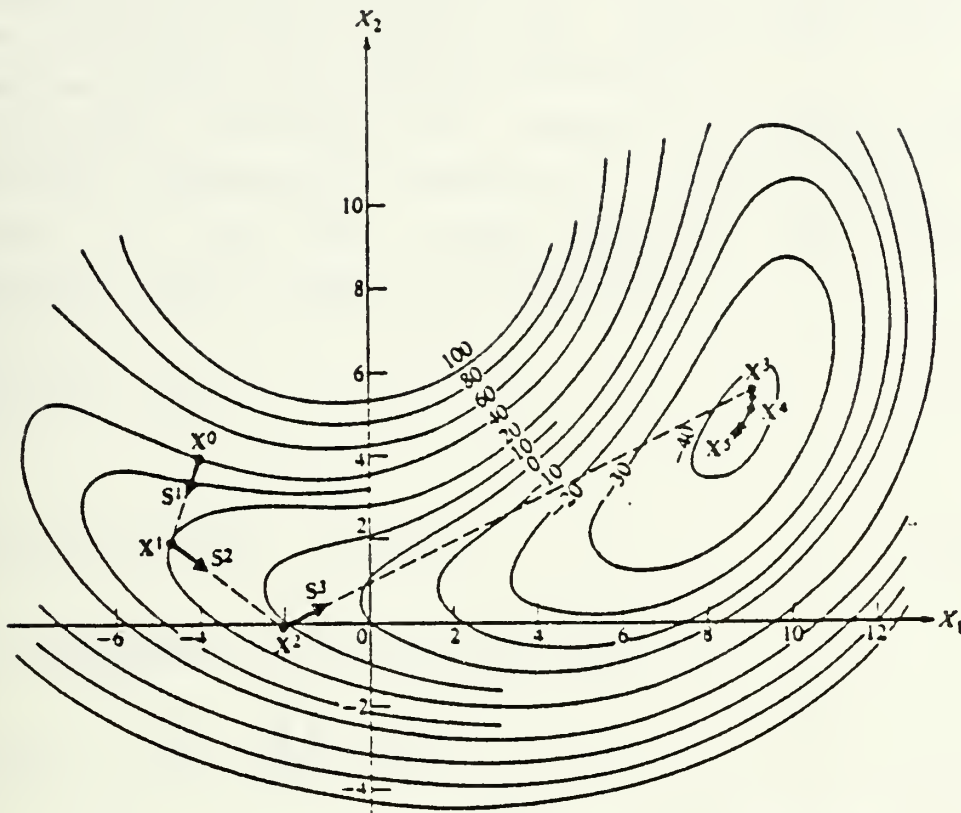


Figure 4.9 Fletcher-Reeves Conjugate Direction Algorithm.

From this equation it is obvious that $P(\underline{x})$ is zero if all the constraints are satisfied. That is, if $g_j(\underline{x}) \leq 0$ and $h_k(\underline{x}) = 0$, then all conditions of the penalty function are satisfied. If an element of the penalty function is violated then the penalty increases as the square of the violated constraint. A pseudo or augmented objective function is formulated where :

$$\Phi(\underline{x}, r_p) = F(\underline{x}) + r_p P(\underline{x}) \quad (4.29)$$

The constant r_p is a weighting factor for the penalty. It is adjusted with ADS as the optimization proceeds to allow the program to systematically converge to an optimum solution.

One disadvantage of this method is that if the optimizer is stopped short of the optimum the design will be in the infeasible region and probably not useful.

Originally developed as a method to solve equality constrained problems the version of ALM in ADS has been modified to work with both equality and inequality constraints. The statement is as follows:

$$\text{minimize } F(\underline{x}) \quad (4.30)$$

$$\text{subject to } h_k(\underline{x}) = 0 \quad k=1,1 \quad (4.31)$$

Next a Lagrangian is created such that

$$L(\underline{x}, \lambda) = F(\underline{x}) + \sum_{k=1}^l \lambda_k h_k(\underline{x}) \quad (4.32)$$

Then the equation is augmented with an exterior penalty function such that

$$A(\underline{x}, \lambda, r_p) = F(\underline{x}) + \sum_{k=1}^l (\lambda_k h_k(\underline{x}) + r_p (h_k(\underline{x}))^2) \quad (4.33)$$

The power of this method is that in theory precise constraint matching is possible whereas in the exterior penalty function method it is not. The full details of the method will not be covered here, however, the final form of the objective function will be included:

$$A(\underline{x}, \lambda, r_p) = F(\underline{x}) + \sum_{j=1}^m (\lambda_j \psi_j + r_p \psi_j^2) + \sum_{k=1}^l (\lambda_k h_k(\underline{x}) + r_p (h_k(\underline{x}))^2) \quad (4.34)$$

where $\psi_j = \max(g_j(\underline{x}), -\lambda_j/2r_p)$, and the update formulas for the Lagrange multipliers are:

$$\lambda_j^{p+1} = \lambda_j^p + 2r_p (\max(g_j(x), -\lambda_j/2r_p)) \quad j=1, m \quad (4.35)$$

$$\lambda_{k+m}^{p+1} = \lambda_{k+m}^p + 2r_p h_k(x) \quad k=1, l \quad (4.36)$$

With this brief review of optimization and the ADS program completed the next chapter will discuss the program development for this thesis research.

V. OPTIMIZATION DESIGN PROCEDURE

To accomplish the task of designing a control system with acceptable time domain performance and robustness characteristics in a straight forward manner a numerical optimization technique was chosen as the method of implementation of the design algorithm. Using numerical procedures to adjust selected design variables, in this case the feedback and/or filter gains, a desired level of performance can be achieved. This level of performance is actually a combination of time domain performance and robustness or frequency domain performance. By establishing the criteria for the system performance in terms of an optimization objective and constraint functions a versatile procedure can be developed to set the system feedback gains and affect an acceptable design in terms of performance and robustness. The pole placement and robustness (POPLAR) design procedure uses pole placement to establish a designer selected performance level and then a minimum singular value level to establish the robustness.

The pole placement portion of the procedure will be discussed first. The pole placement technique was chosen because it was relatively easy to implement through a numerical optimization routine. By using this numerical procedure it is also simple to incorporate robustness into the procedure along with the performance requirements. A numerical technique similar to one posed in [Ref. 13] was chosen for the pole placement algorithm. An unconstrained optimization routine from the IBM IMSL library was used for this program. The routine, a Newton method, uses adjustments to the output feedback gains to reduce the size of an objective function. This objective function was expressed as a function of the

pole location of the system and as the objective was reduced the poles were moved toward the desired locations. To provide more versatility in the pole placement algorithm a method that can use constraint functions as well as unconstrained optimization was chosen for this program. The designer may use either objective, constraint or a combination of functions to secure the desired pole locations. As currently implemented in the program the cost or objective portion of the pole placement procedure is constructed as equation 5.1

$$OBJ = \sum_{i=1}^J (\lambda_{R_{D_i}} - \lambda_{R_i})^2 + (\lambda_{I_{D_i}} - \lambda_{I_i})^2 \quad (5.1)$$

where λ_R = real eigenvalue
 λ_I = imaginary eigenvalue
 λ_{R_D} = desired eigenvalue location
 λ_{I_D} = desired eigenvalue location

The constraint formulation is a function that must be kept negative or the constraint is violated. It is written as equation 5.2

$$g(j) = \sqrt{(\lambda_{R_{D_j}} - \lambda_{R_j})^2 + (\lambda_{I_{D_j}} - \lambda_{I_j})^2} - r \quad (5.2)$$

where r is a tolerance circle established as a function of pole placement position. Since the aim of the optimizer is to keep g negative any time the λ function of the constraint is greater than r the constraint will become active, i.e. violated. The optimizer will then attempt to move the constraint to the inactive status by adjusting the design parameters of the system.

Consideration of implementation of the frequency domain or robustness portion of the design procedure begins with the concept of MIMO phase and gain margins. Several useful theorems on singular value analysis of multiloop systems are

presented in [Ref. 8]. One of these theorems relates the matrix singular value of the return difference function to a parameter, α , and further shows that as long as the maximum singular value of the perturbation function ($\underline{I}^{-1} - \underline{I}$) remains less than this α , the system remains stable. The value of α is then related to gain and phase margins of the MIMO system. The relationship developed is given in equations 5.3 and 5.4:

$$\text{gain margin} = \text{GM} = 1/(1+\alpha_0) \quad (5.3)$$

$$\text{phase margin} = \text{PM} = \pm \cos^{-1}(1-\alpha_0^2/2) \quad (5.4)$$

provided that equation 5.5 holds.

$$\sigma(\underline{I}+\underline{G}) \geq \alpha_0 \quad (5.5)$$

for some $\alpha_0 \leq 1$

These phase and gain margins are guaranteed in every loop simultaneously.

Universal gain and phase margin curves, [Ref. 14], based on the minimum singular values of the return difference matrix are developed from equation 5.6.

$$(\underline{I}^{-1}-\underline{I}) = \max \sqrt{(1-1/k_n)^2 + 2/k_n(1-\cos \phi_n)} \quad (5.6)$$

for all n with $k_n > 0$. These curves shown in figure 5.1 allow the designer to pick a singular value that corresponds to a specific gain and phase margin for a given system. In addition to the universal gain and phase plot [Ref. 15] develops an optimizer solution for formulating a robust controller using the CONMIN algorithm [Ref. 12].

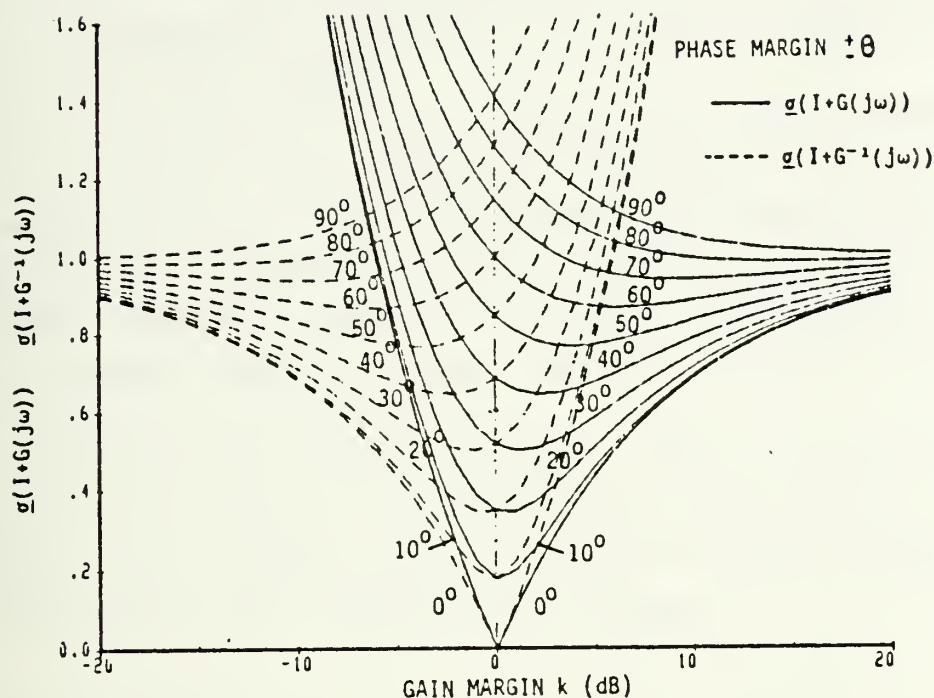


Figure 5.1 Universal Gain and Phase Singular Value Plot.

Since the universal curve in figure 5.1 provides a convenient method of specifying gain and phase margins in terms of singular values the robustness portion of the pole placement and robustness design procedure uses the minimum singular value level of the return difference matrix to determine the robustness. The minimum singular value level is formulated as an objective or constraint function in equation 5.7

$$J = \sum_j (\max(0, (\sigma_p - \sigma(j\omega, p)))^2 \quad (5.7)$$

The optimization procedure may be used to change feedback gains until the minimum singular value is raised above this desired design level. Although the same formulation can be used as a negative constraint function it has not been implemented as such within this program. There are numerous

ways the singular value formulation could be implemented within the program by changes of the code if design requirements forced such changes.

The pole placement and robustness design program is based on the ADS code discussed in Chapter 4 to implement the design variable selection procedures. The pole placement and robustness program consists of two separate programs. The first program is used to provide designs for state or output feedback problems while the second program is used for observer or filter designs.

For the state or output feedback design program the user must input the plant matrices \underline{A} , \underline{B} , \underline{C} and initial starting values for the feedback matrix \underline{F} . The matrices correspond to the following linear differential system:

$$\dot{\underline{x}} = \underline{A}\underline{x} + \underline{B}\underline{u} \quad (5.8)$$

$$\underline{y} = \underline{C}\underline{x} \quad (5.9)$$

$$\underline{u} = -\underline{F}\underline{x} \quad (5.10)$$

A feed-forward matrix has not been considered in the current program. A feed forward matrix could be added to the procedure if required for specific design cases.

As the design program is currently coded the user may run output feedback or state feedback design by specifying the \underline{C} matrix as the diagonal(\underline{I}) matrix for state feedback. The program relies on initial starting values of the feedback gains, \underline{F} . As discussed in Chapter 4 there is no guarantee that the optimum found by the procedure each time is the global optimum or that the procedure will always

converge to an acceptable solution. The ability to select acceptable starting values for the feedback gains will make the procedure more efficient in operation. As currently employed, the program is used to obtain pole placement and robustness for a given set of starting gains and a selected optimization routine from the ADS program. If the optimizer is not able to meet the desired design goals on this program run two options are available. First, change to a different optimization routine from the list of available ADS routines and rerun the problem. This was usually successful in improving the design. Second, the designer uses a new set of starting values for the feedback gains and repeats the design procedure. Both options might be used on particularly difficult cases.

The pole placement and robustness design procedure has consistently been able to find improved designs; however the program does not always yield acceptable designs. Certain problems require changes in the optimizer routine and modification in the initial feedback gain starting values in order to obtain acceptable designs. Using the IBM 3033 time share system the pole placement and robustness routine requires about ten CPU seconds to work a second order problem and on the order of 15 to 60 CPU seconds to run a fourth order problem. The actual amount of time varies with optimization requirements and time share utilization.

The observer robustness design program requires two passes of the ADS program. In the first pass the feedback gains, F , of the controller are computed to obtain the desired pole locations. The second pass of the ADS routine is used to adjust the observer gains to recover the system robustness. The two pass procedure was chosen because it allows a smaller number of design variables at each stage of the optimization and much more efficient computer usage. Figure 5.2 shows how the observer is implemented. This diagram is algebraically stated as;

$$\begin{aligned} \dot{\underline{x}} &= \begin{bmatrix} A & -BK \\ FC & A_c - BK \end{bmatrix} \begin{bmatrix} \underline{x} \\ \hat{\underline{x}} \end{bmatrix} + \begin{bmatrix} B \\ B \end{bmatrix} r \\ \hat{\underline{x}} &= A_c \underline{x} - FC \end{aligned} \quad (5.11)$$

where \underline{x} is the state, $\hat{\underline{x}}$ is the estimator variable, F is the feedback gain and K is the observer gain. The design of the feedback gains and the observer gains are accomplished as separate quantities in keeping with the separation prin-

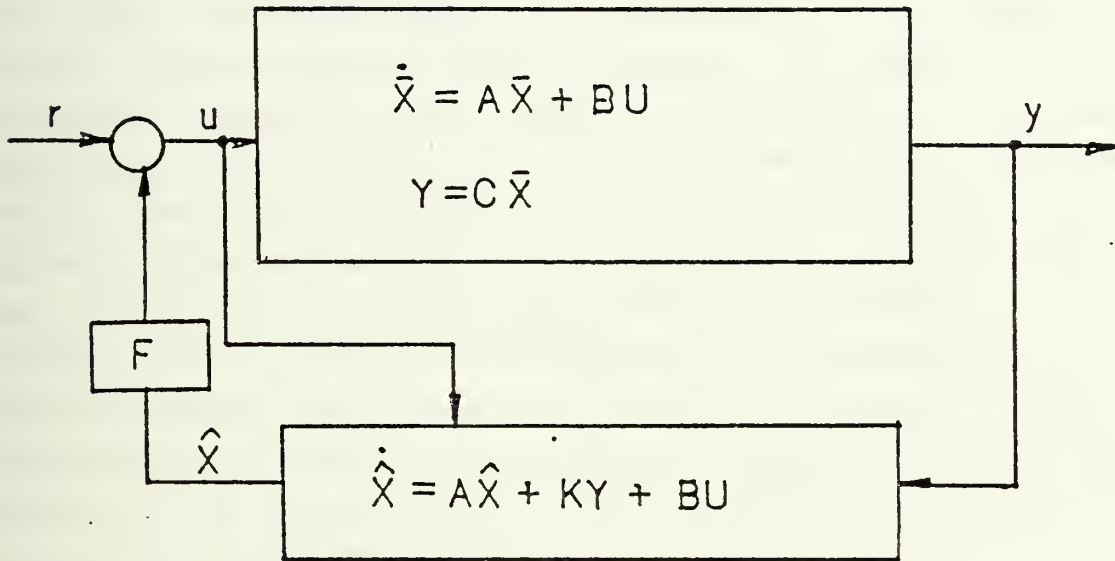


Figure 5.2 Observer Implementation.

ciple. In using the pole placement and robustness design procedure for the observer system, initial values of the F and K matrices must be input. The same or different optimization techniques from ADS may be employed.

The routines contained in this thesis have been based on the input additive singular value level. The pole placement and robustness design algorithm computes input additive,

output additive, input multiplicative, and output multiplicative singular values. Any of these singular values can be incorporated into the objective or constraint formulations but have not been for this version of the program.

In summary, the pole placement and robustness design procedure is a straight forward numerical optimization procedure for the practical application modern MIMO system analysis. The new aspects of the procedure are the implementation of both pole placement and robustness criteria within the same design program. The versatility of the pole placement and robustness design is obtained by incorporating a state of the art optimizer routine ADS, with currently available singular value computation routines. Using the optimizer format for the pole placement and robustness design gives the designer the ability to modify variables directly that affect both the time domain or performance of the system and the frequency domain or robustness of the system. The numerical optimization incorporated into the pole placement and robustness design is flexible enough to incorporate the double pass design technique using the separation of the feedback and filter gains to obtain robustness recovery for observer based controllers.

VI. INTRODUCTORY PROBLEM

The purpose of this introductory problem is to review some results of modern multivariable robustness theory using a simple problem. This same simple system is then used as a test problem for the optimization technique developed for this thesis. The problem provides excellent insight into the cross-coupling problem and demonstrates how effectively the pole placement and robustness design procedure can be in dealing with the cross perturbation terms.

The problem chosen for this introductory analysis comes from [Ref. 9]. Figure 6.1 is a diagram of this basic system. In this problem a simple plant is specified by the following linear system:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 1 & b_{12} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \quad (6.1)$$

where

$$y_1 = x_1 \quad (6.2)$$

$$y_2 = x_2$$

A feedback compensation of the form of equation 6.3 was assumed.

$$\begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = - \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} uc_1 \\ uc_2 \end{bmatrix} \quad (6.3)$$

Which gives a closed-loop system, equation 6.4

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} -2 & b_{12} \\ 0 & -2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 1 & b_{12} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} uc_1 \\ uc_2 \end{bmatrix} \quad (6.4)$$

This system has eigenvalues at $-2, -2$ and is therefore stable. Using equation 6.5

$$\underline{G} = \underline{C}(\underline{sI} - \underline{A})^{-1}\underline{B} \quad (6.5)$$

the transfer matrix may be written as equation 6.6.

$$\underline{G} = \begin{bmatrix} 1/s+1 & b_{12}/s+1 \\ 0 & 1/s+1 \end{bmatrix} \quad (6.6)$$

which gives the return difference matrix, equation 6.7.

$$\underline{I} + \underline{G}(s) = \begin{bmatrix} s+2/s+1 & b_{12}/s+1 \\ 0 & s+2/s+1 \end{bmatrix} \quad (6.7)$$

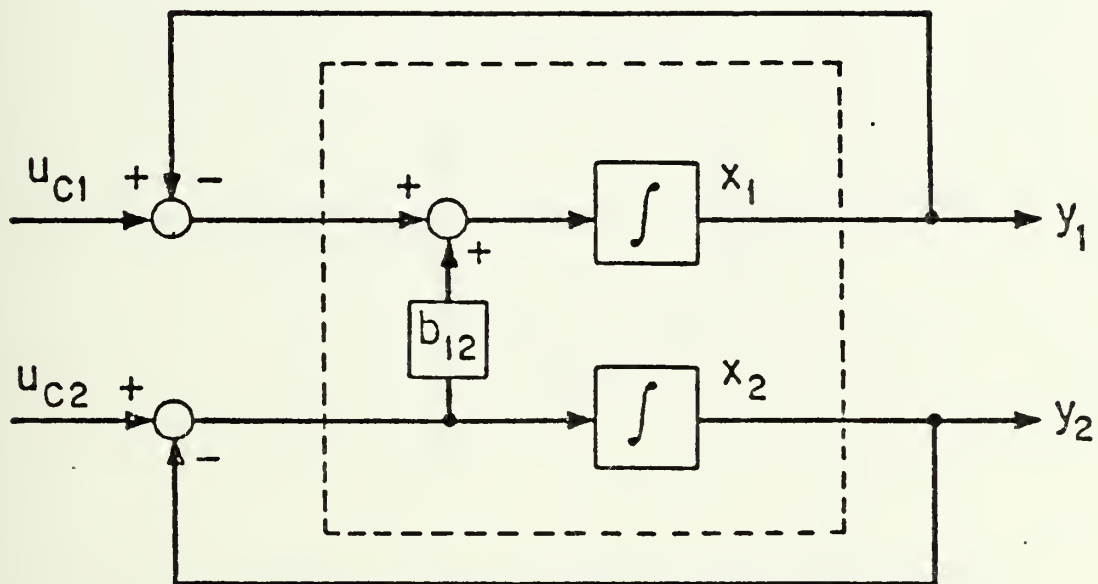


Figure 6.1 Basic Multi-input Multi-output System.

The problem shows the inadequacies of classical methods in establishing the robustness of the system. A brief review

of the results will be presented. Using the return difference matrix $(\underline{I}+\underline{G})$, the determinant may be written as equation 6.8.

$$\det(\underline{I}+\underline{G}) - 1 = 2s+3/(s+1)^2 \quad (6.8)$$

Therefore, the multivariable Nyquist diagram will be as

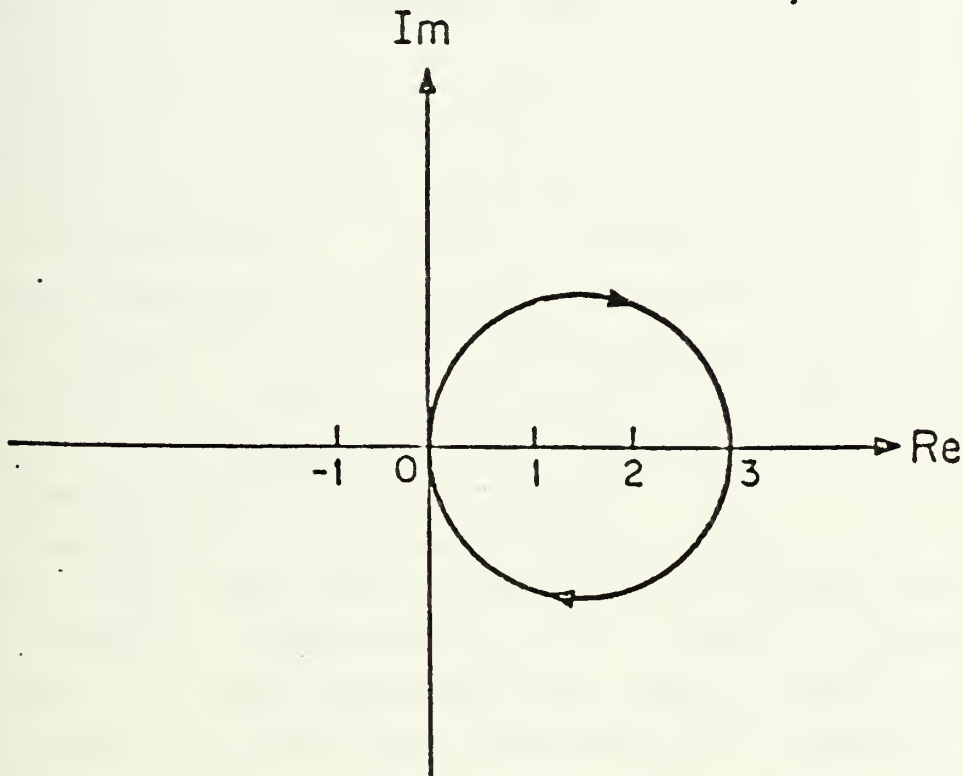


Figure 6.2 Multivariable Nyquist for $2s+3/(s+1)^2$.

shown in figure 6.2. The diagram does not encircle the $(-1,0)$ point and is indicative of a closed-loop stable system. Considered as a SISO system one has a gain margin of $-1/3$ to ∞ and a phase margin of ± 106 degrees. Under this criteria one can conclude that this is a good design. This will be shown later not to be the case.

Multi-input systems are often designed a loop at a time. Applying loop design to the system the transfer function for either loop taken separately becomes equation 6.9

$$G=1/s+1$$

(6.9)

Figure 6.4 shows the Nyquist diagram of this problem. Using this Nyquist diagram the system is indicated to be stable and have phase and gain margins of $GM=(-1,\infty)$ and $PM=(\pm 180)$. This analysis does not show the true nature of the robustness of the system. Since the factor b is not a parameter in either of the Nyquist curves it plays no part in the stability determination using these diagrams.

Using the criteria of singular values discussed in Chapter 3 a measure of the nearness to instability for this problem may be obtained by plotting the minimum singular value of $(\underline{I}+\underline{G}), \underline{G}$. (For numerical calculations a value of $b_{12} = 50$ is assumed). Figure 6.3 shows the plot of this value vs. frequency. This gives a minimum singular value of about -23 db or 0.071 which corresponds to a gain margin of about 0.93 to 1.08 and a phase margin of ± 4.1 degrees. These phase and gain margins are quite small and are evident in the cross-feed perturbation problem.

A perturbed system as shown in figure 6.5 can be produced which will lead to stability problems with very small values of perturbation. The closed-loop system if perturbed by a small perturbation, $5/b_{12}$, where b_{12} is a large number, will have as a characteristic equation, equation 6.10

$$(s\underline{I}-\underline{A})= s^2 + 4s + 9 \quad (6.10)$$

with the eigenvalues of $s=-2\pm\sqrt{5}$. There is one positive root in this solution and the system is unstable.

To determine the nature of the robustness of this system the return difference matrix must be considered. If the return difference matrix of the transfer function $(\underline{I}+\underline{G}(j\omega))$

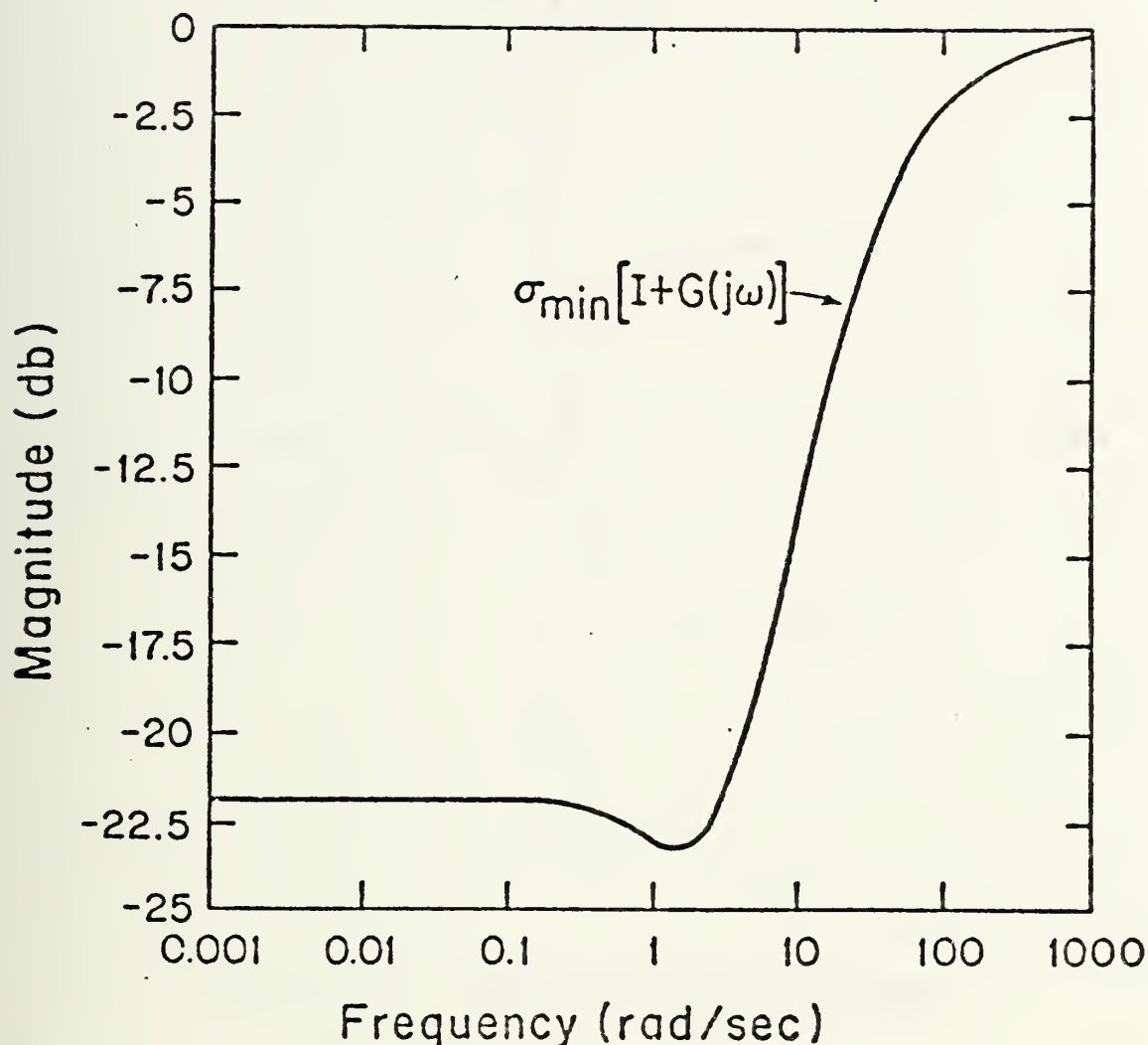


Figure 6.3 Minimum Singular Value Plot, Example Problem.

is nearly singular at some frequency ω_0 , then the multivariable system will not be robust with regard to any modelling errors within the system. This is because any small change in $G(j\omega)$ can then make $I+G(j\omega)$ singular and the $\det(I+G)$ becomes zero, thus changing the encirclements of the Nyquist stability point and indicating a system instability. Using equation 6.1 the pole placement and robustness design method

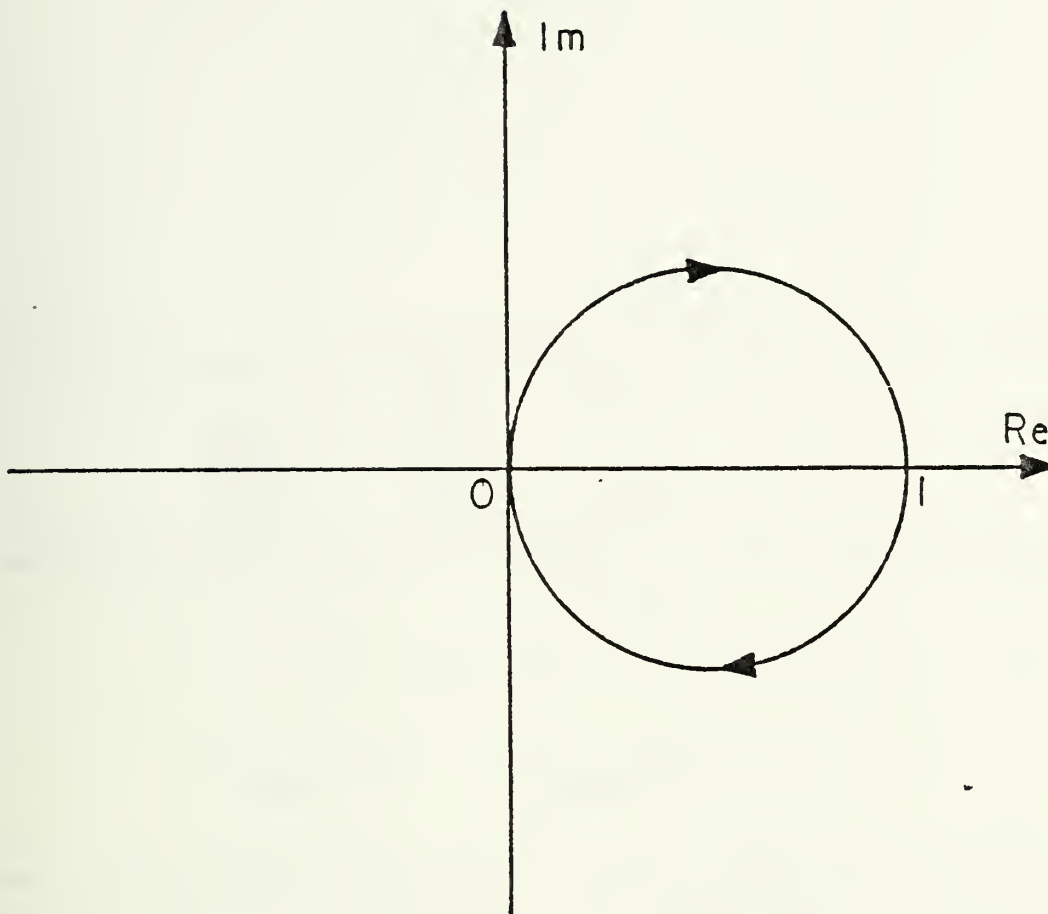


Figure 6.4 Nyquist Diagram of $1/s+1$.

can be demonstrated . The first step is to establish a baseline for the design. The state unity feedback model in this simple problem gives pole locations of -2 and -2 . This set of eigenvalues -2 , -2 were chosen as the baseline for the system. Since a second order system requires only two feedback gains to place the poles, the diagonal feedback gains were chosen for pole placement purposes. The pole placement and robustness design program was then used to place the poles and in the process obtained the gains required to do this (1,1). A plot of some of the singular value criteria obtained is shown in figure 6.6. Pole-zero plots of the

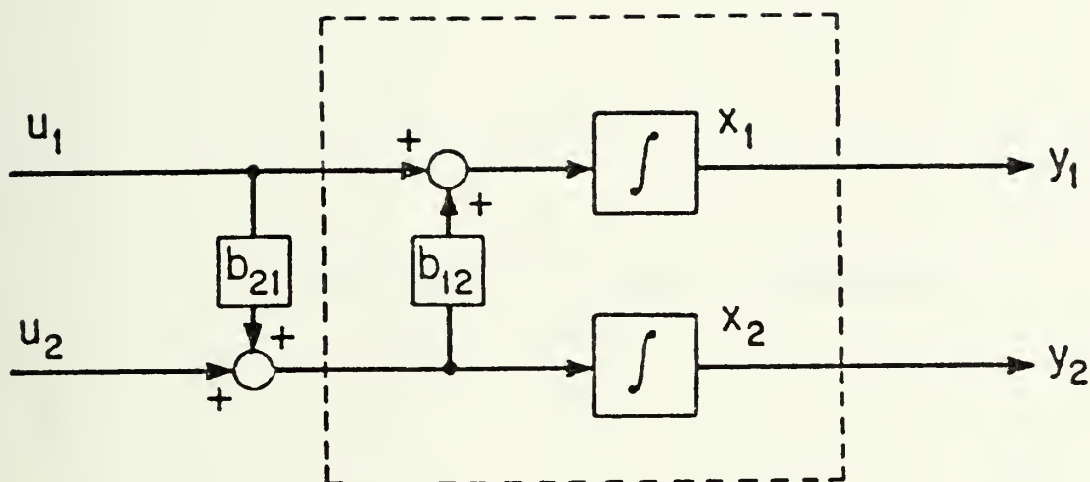


Figure 6.5 Perturbed System.

closed-loop transfer functions of the closed-loop transfer matrix are shown in figure 6.7. In this case the poles and a zero are clustered about the -2 point and in the input two to output one channel a zero is located at the -1 point on the pole-zero diagram. This point closely corresponds to the minimum singular value frequency. The only significant observations are the ability of the pole placement and robustness routine to place the poles and the relatively poor singular values indicative of low robustness.

Since there is a requirement for two feedback parameters to set the poles of a second order system there should be no additional freedom in design to account for robustness. Case two was a run to demonstrate this fact. Again allowing only two design variables for the problem the pole placement and robustness design program was run but with an objective function formulated to adjust the singular value above a

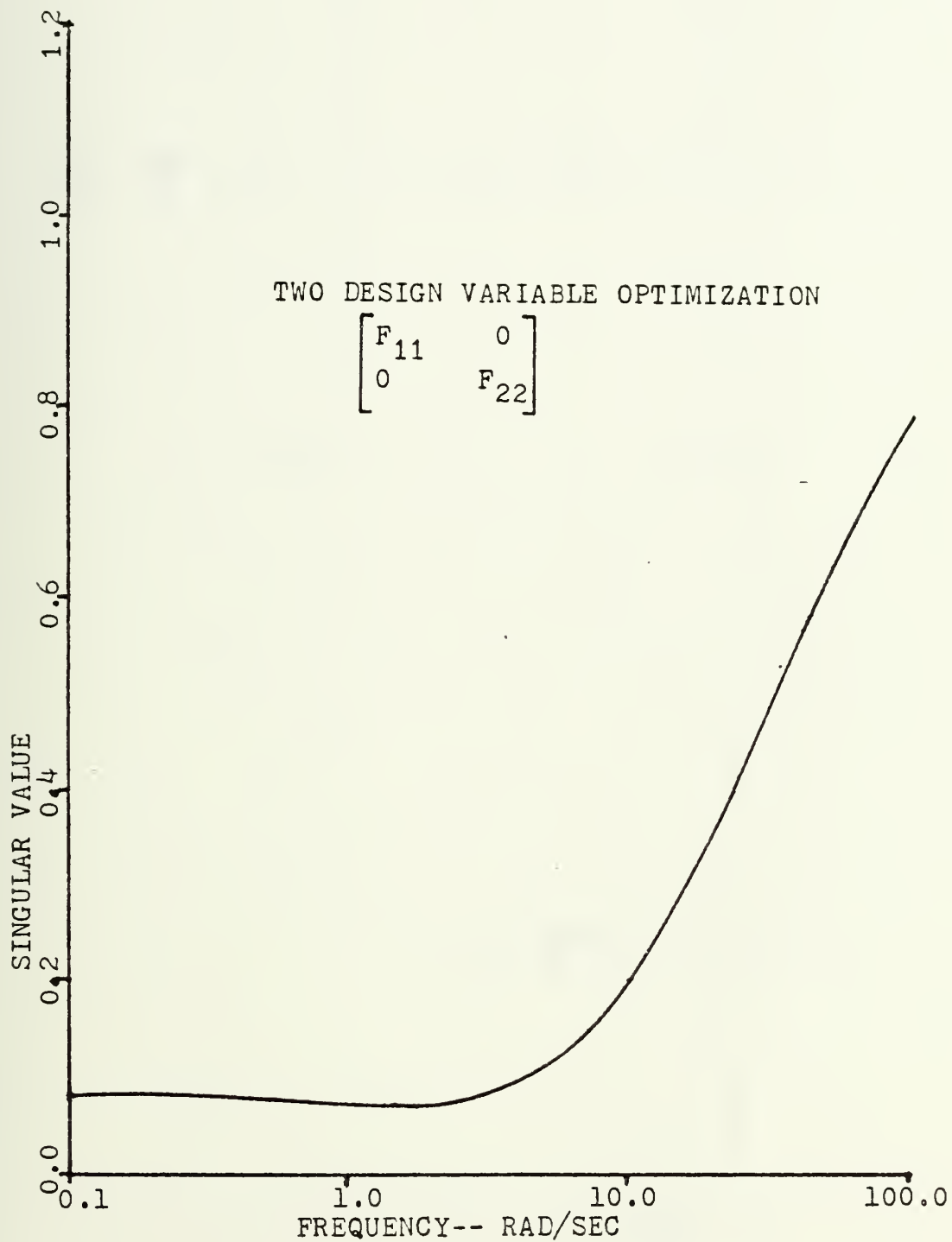


Figure 6.6 Singular Value Plot for Simple Problem.

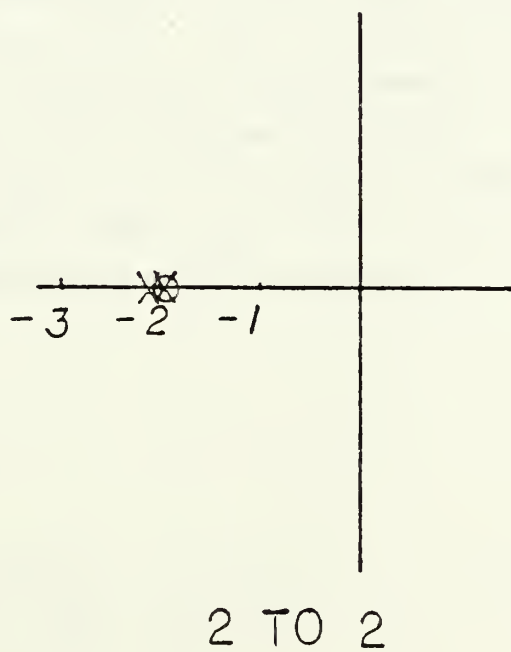
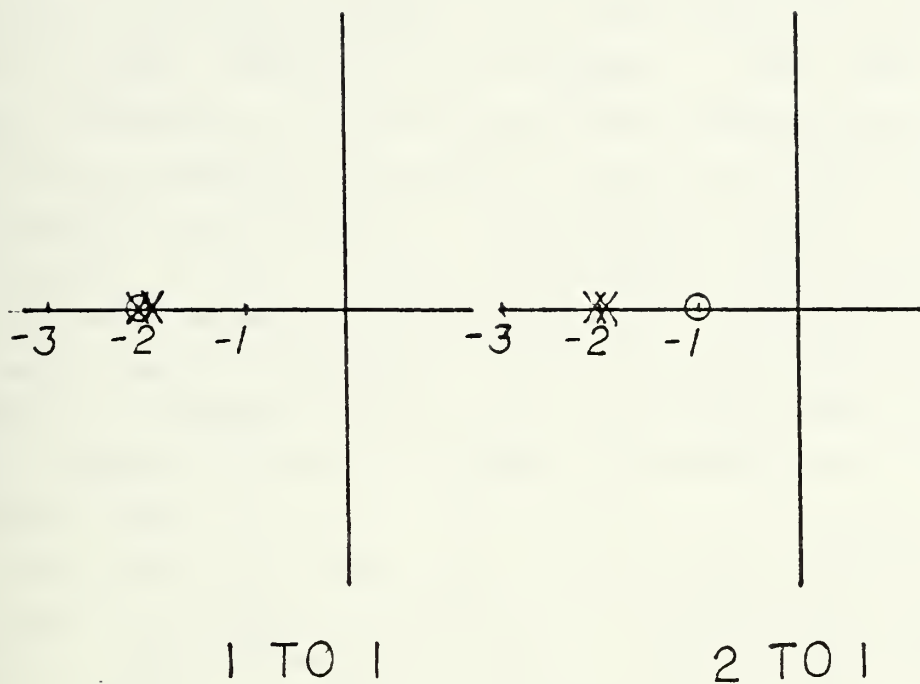


Figure 6.7 Closed-loop Poles for Pole Placement Only Case.

base level of 0.6. This level would correspond, using the universal gain margin chart, to a gain margin of -4 db to 8 db and a phase margin of ± 35 degrees. With only the two design variables to work with the pole placement and robustness design program was unable to place the poles and adjust the singular value level to the required value. The poles were placed at -2, -2 but the singular value minimum was still on the order of -23 db or 0.07. The pole-zero plot remained almost unchanged. It is clear that additional degrees of freedom for the pole placement and robustness design program must be opened if robustness is to be accounted for.

Case three was then run on the pole placement and robustness program by adding on additional degree of freedom. This case used the f gain as the additional design variable. A good choice as will be seen. Allowing the optimizer routine the extra freedom to adjust the additional feedback gain term an excellent design was found. The singular value minimum became 0.885 as shown in figure 6.8. Using the universal gain margin chart this corresponds to gain and phase margins of -6 db - 18 db and ± 52 degrees respectively. This is a considerable improvement over the original design. The factor that changed the design was the upper diagonal feedback term which provides a cancelling factor for the cross-coupling term $b_{12} = 50$. This can be seen by looking at the system matrix equation 6.11.

$$\underline{A}_c = \underline{A} - \underline{B}\underline{F} \quad (6.11)$$

where equation 6.12 gives \underline{A} .

$$\underline{A}_c = \begin{bmatrix} -1-f_{11} & -f_{12} - b_{12} f_{22} \\ 0 & -1-f_{22} \end{bmatrix} \quad (6.12)$$

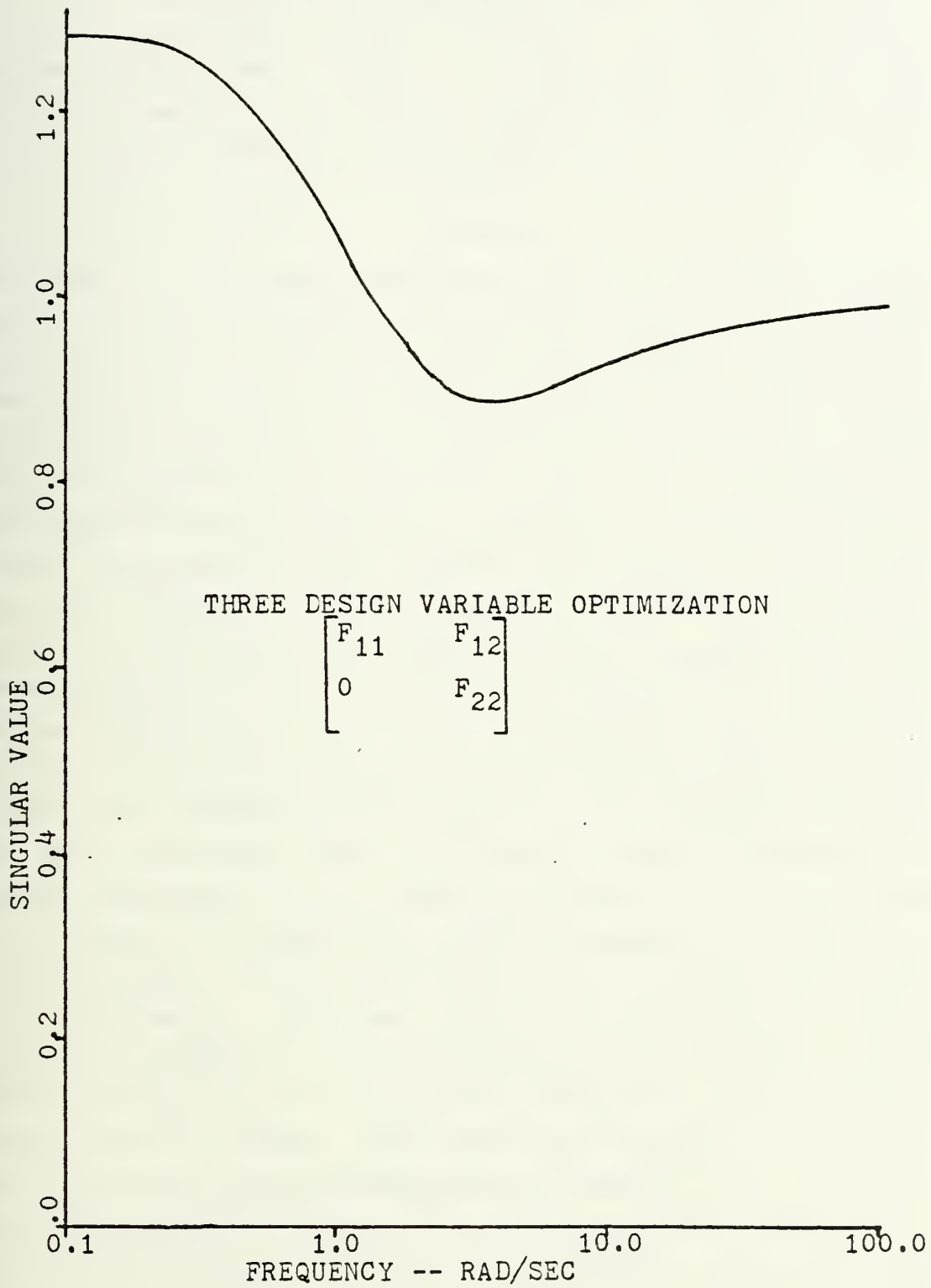


Figure 6.8 Singular Value Plot Case Three.

If the upper diagonal term of the matrix can be driven to zero the system will be decoupled into a diagonal system. To place the poles of this system at $-2, -2$ the diagonal feedback gains must be one. The optimizer feedback gains for this case were $f_{11} = 1.00006$, $f_{22} = 0.99998$ and, most importantly, $f_{12} = -51.83401$. Using these values in the A matrix

$$A = \begin{bmatrix} -2.00006 & 1.83501 \\ 0.0 & -1.99998 \end{bmatrix}$$

The value in the upper right position in the system matrix has been lowered considerably from the value of near 50 that appears in this position in the low singular value cases. Lowering this system gain value decreases the cross-coupling perturbation effects on the system. This can be more graphically demonstrated by figures 6.9 and 6.10. In figure 6.9 the transfer function shows a high gain of approximately 35 db and a bandwidth of 50 rad/sec. In figure 6.10 this gain has been reduced to 6.0 db with a bandwidth of almost 2 rad/sec. Two things are indicated by the figures, one, the open-loop Bode plot of the transfer function of the cross-coupled channel can be used to indicate the robustness problem as evidenced by the high gain and bandwidth relative to the other feedback gains and, two, the mechanism used by the pole placement and robustness design procedure to recover robustness is to reduce the relative gain and associated bandwidth within the affected channel. Figures 6.11 and 6.12 which are for the input one to output one channel are approximately the same as are the Bode plots for the input two to output two channel which are not shown. These figures indicate that no problem exists in the diagonal or direct coupling terms. The pole-zero diagram for optimization run case three is included in figure 6.13. The only significant change in this plot as compared to figure 6.7 is the movement of the zero in the off diagonal pole-zero plot. The zero is seen to shift to one of the pole locations

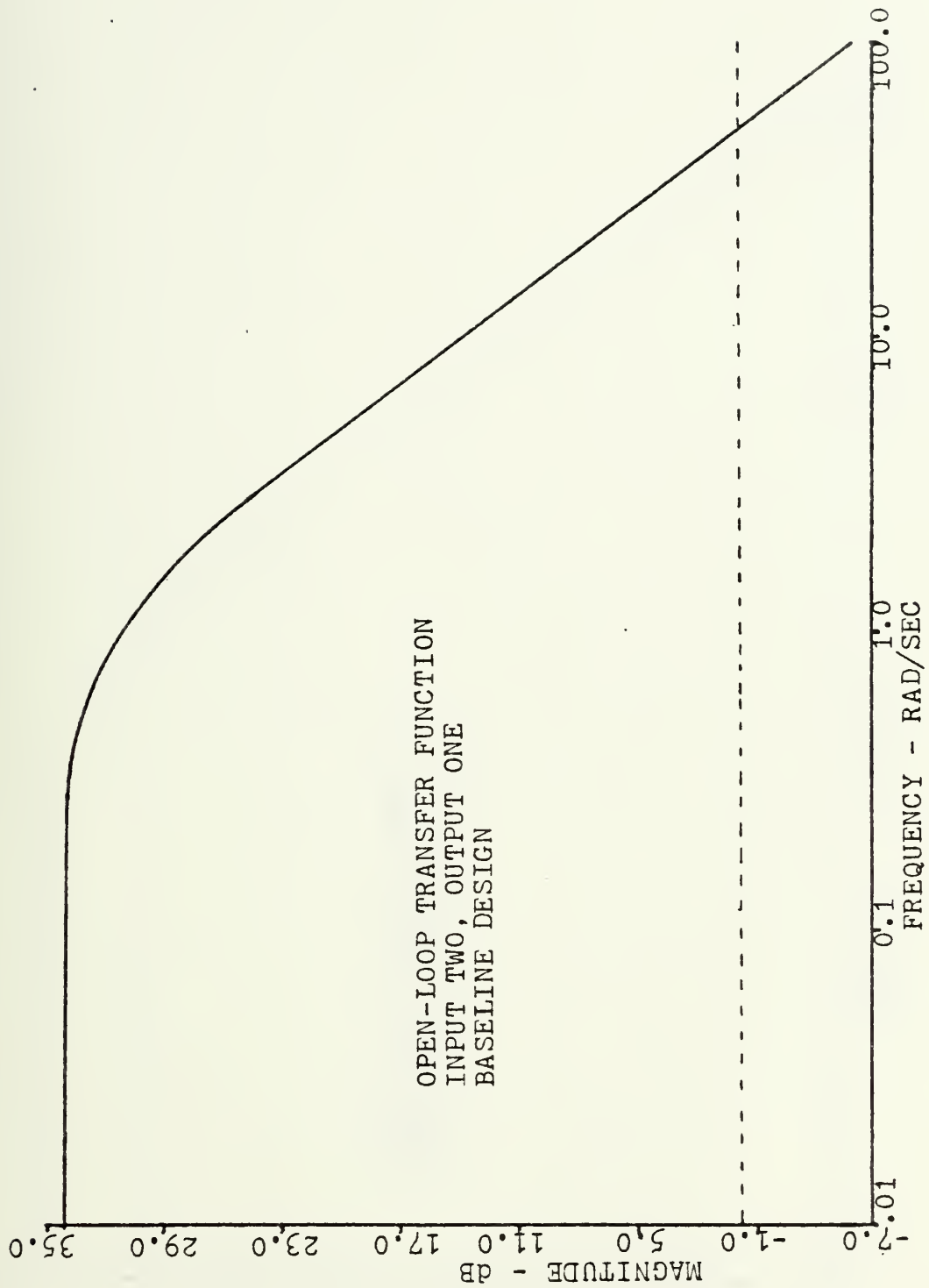


Figure 6.9 Open-loop Transfer Function 2-1 for Baseline.

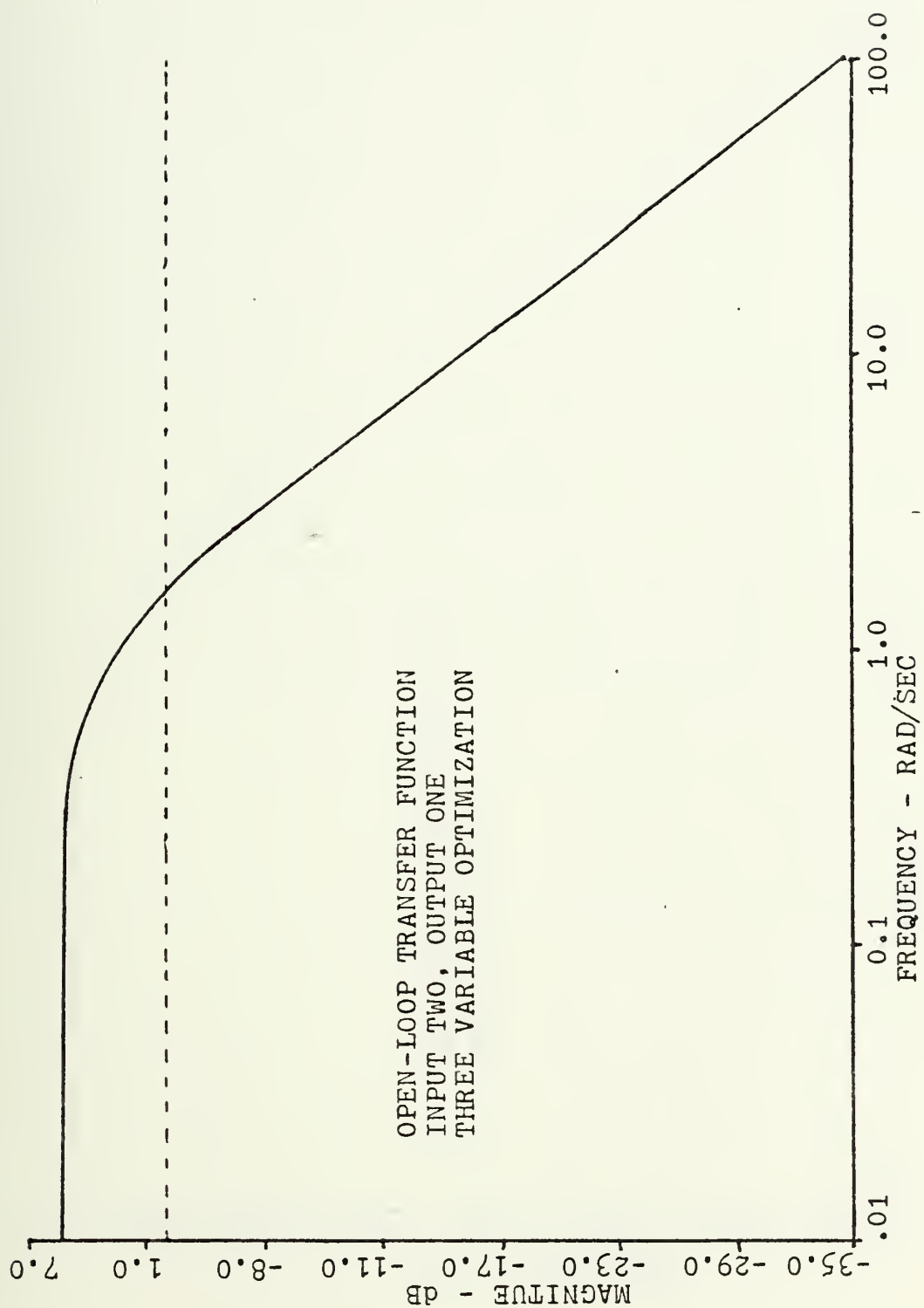


Figure 6.10 Transfer Function 2-1 Optimized (3 Var).

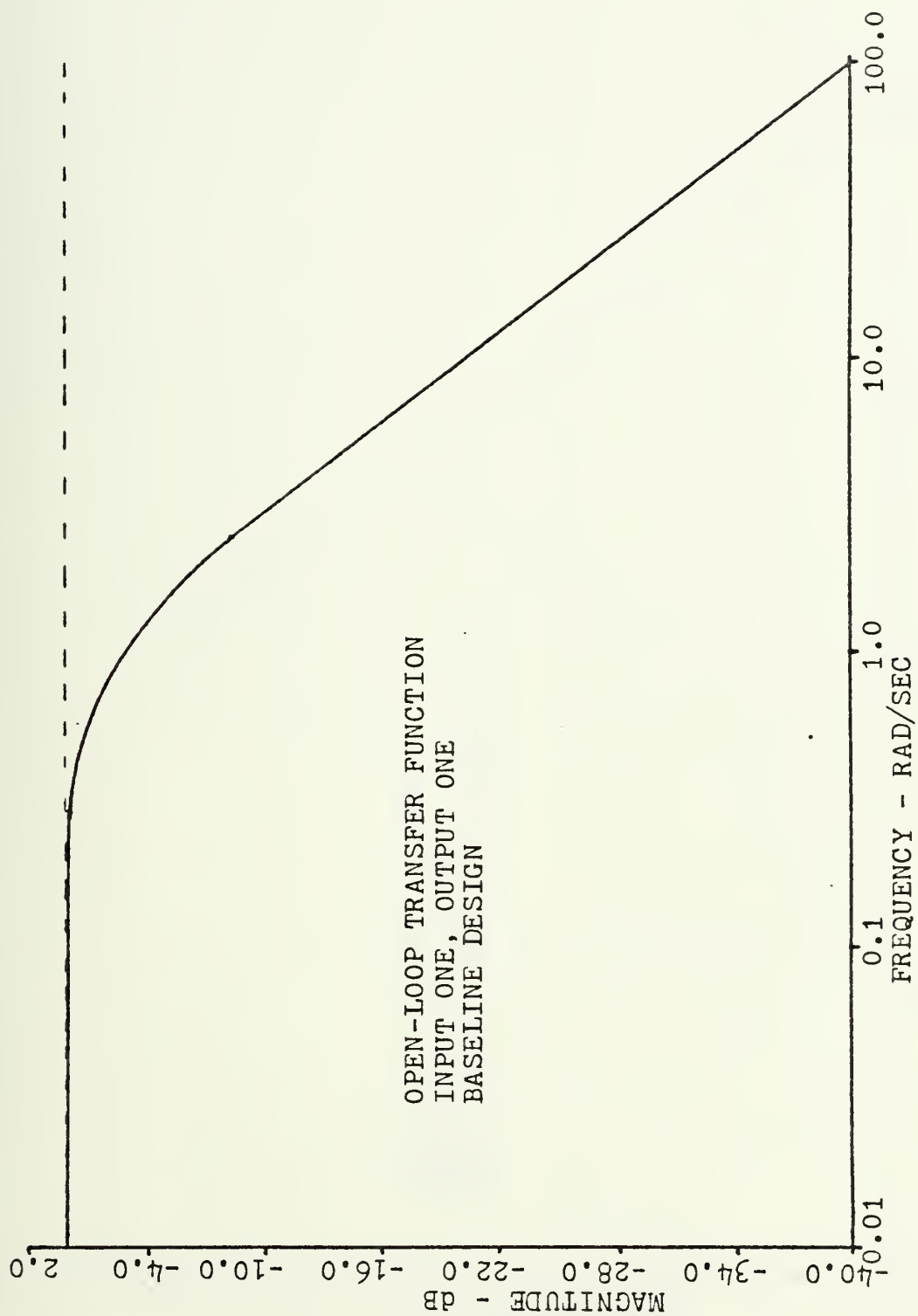


Figure 6.11 Transfer Function 1-1 Baseline.

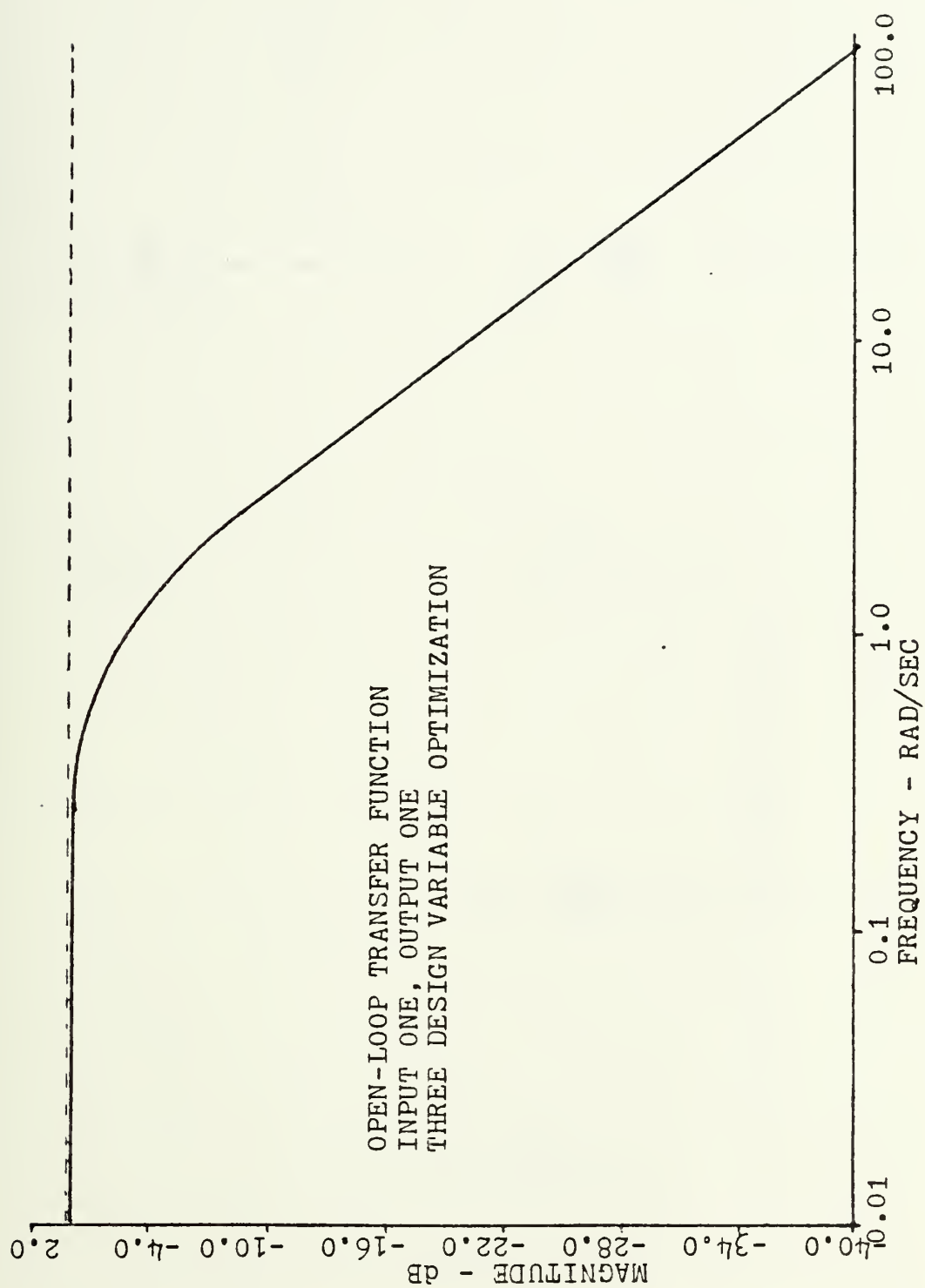


Figure 6.12 Transfer Function 1-1 Optimized (3 variables).

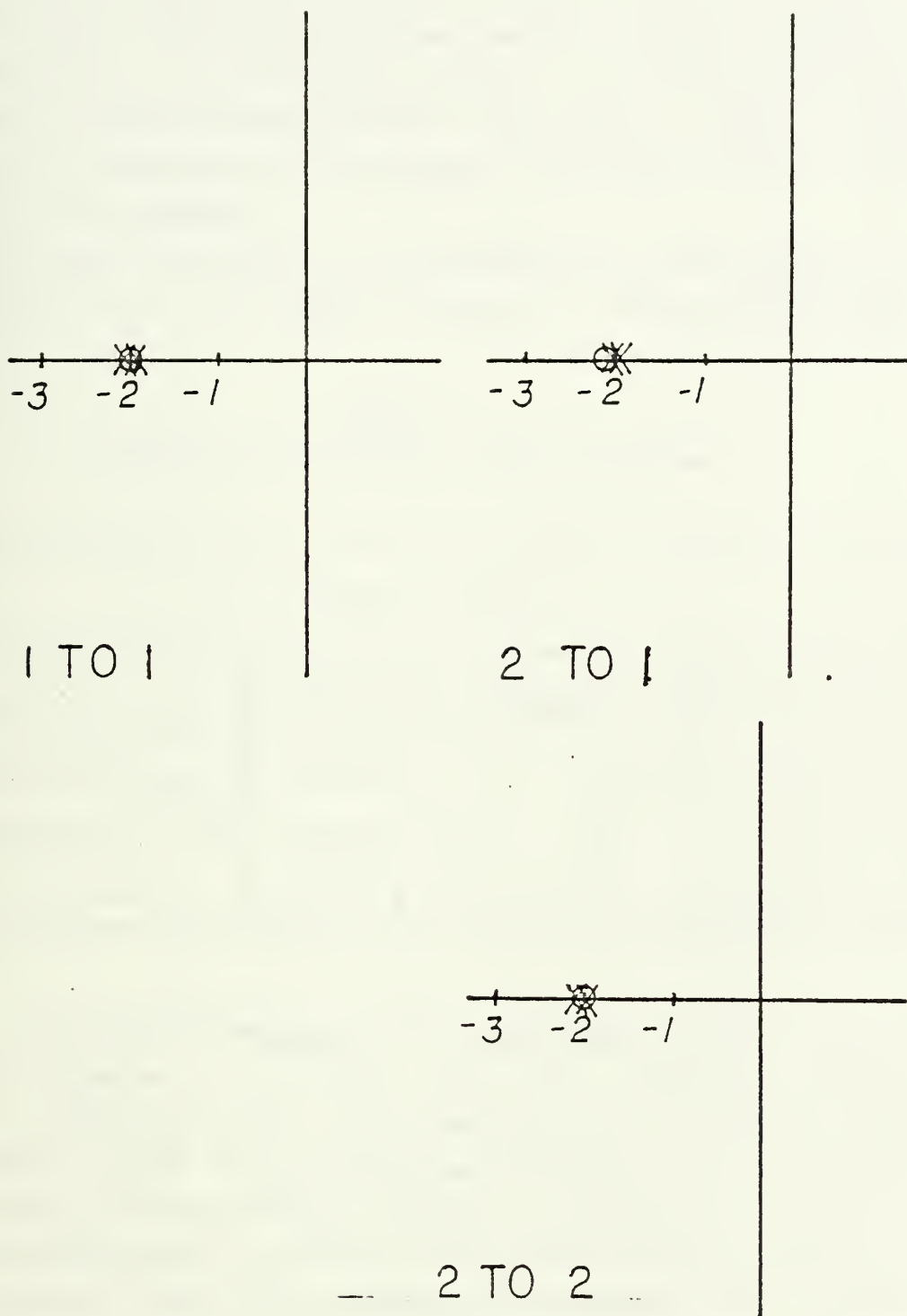


Figure 6.13 Closed-loop Pole-Zero Plot Case 3.

during the optimization. This zero shift has the effect of smoothing the frequency response curve in the vicinity of the frequency of the minimum singular value providing a more uniform gain distribution around this point. Table 1 shows comparison results for the feedback gains for several cases of this basic problem.

The logical extension of the problem to the general case is to allow all four feedback gains to become design vari-

TABLE 1
Comparative Results Simple Problem

	Feedback Gains			
	f_{11}	f_{12}	f_{21}	f_{22}
f_{12} variable $f_{21}=0$	1.00006	-51.83401	0.0	0.9998
f_{21} variable $f_{12}=0$	0.25254	0.0	-0.01071	1.52332
f_{21} variable $f_{12}=-50$.99960	-50.0	0.00025	0.99038
All f's variable	1.11936	-60.4718	0.00280	0.75478

ables. The pole placement and robustness routine can then use full freedom in choosing all four of these feedback gains to compensate for any cross-coupling effects within the system. Based on the previous analysis the two diagonal gains would be expected to approach 1 while the upper off diagonal gain moves to -50 and the lower one moves to 0. The flexibility of the ADS program was required for the four design variable study. Several runs were made with various combinations of starting parameters for the feedback gains and optimizer routines before a good design for the case

incorporating all four feedback gains as variable was obtained. While this points out one of the limitations of optimization design routines the program was able to develop an improved design over the baseline while employing four design variables. In the design produced for this case the singular value level was placed above 0.6 for a gain margin of -4 db to 9 db and phase margin of ± 35 degrees. In this formulation the optimizer was able to place the poles and meet the design singular value level. The feedback gains produced by the optimizer were:

1.11936	-60.4718
0.0028	0.75478

which are approaching the analytic design gains. After obtaining the feedback gains the OPTSYS program was used to obtain the necessary data to do a closed-loop pole-zero map. This plot is shown in figure 6.14. These plots indicate a similar pole-zero location to that found in the previous three design variable problem. Again the gain in the affected channel has been reduced to compensate for the high cross-coupling perturbation within the system.

The design studies presented to this point have been based on breaking the system loop at the input as shown in figure 6.15. In multivariable theory the location of the break in the loop changes the return difference for the system and the transfer function formulation. In the figure number 1 depicts a system with an input loop break point while number 2 is an output loop break point for output return difference determination. The return difference function for the point 1 is written as $I + F_1 G_1$ while the return difference for point 2 is $I + G_2 F_2$. The baseline system not only has low singular values of the input return difference matrix, the lowest being 0.0706, but also has low singular values for the output return difference matrix of point 2 of about the same order. To demonstrate the versatility of the

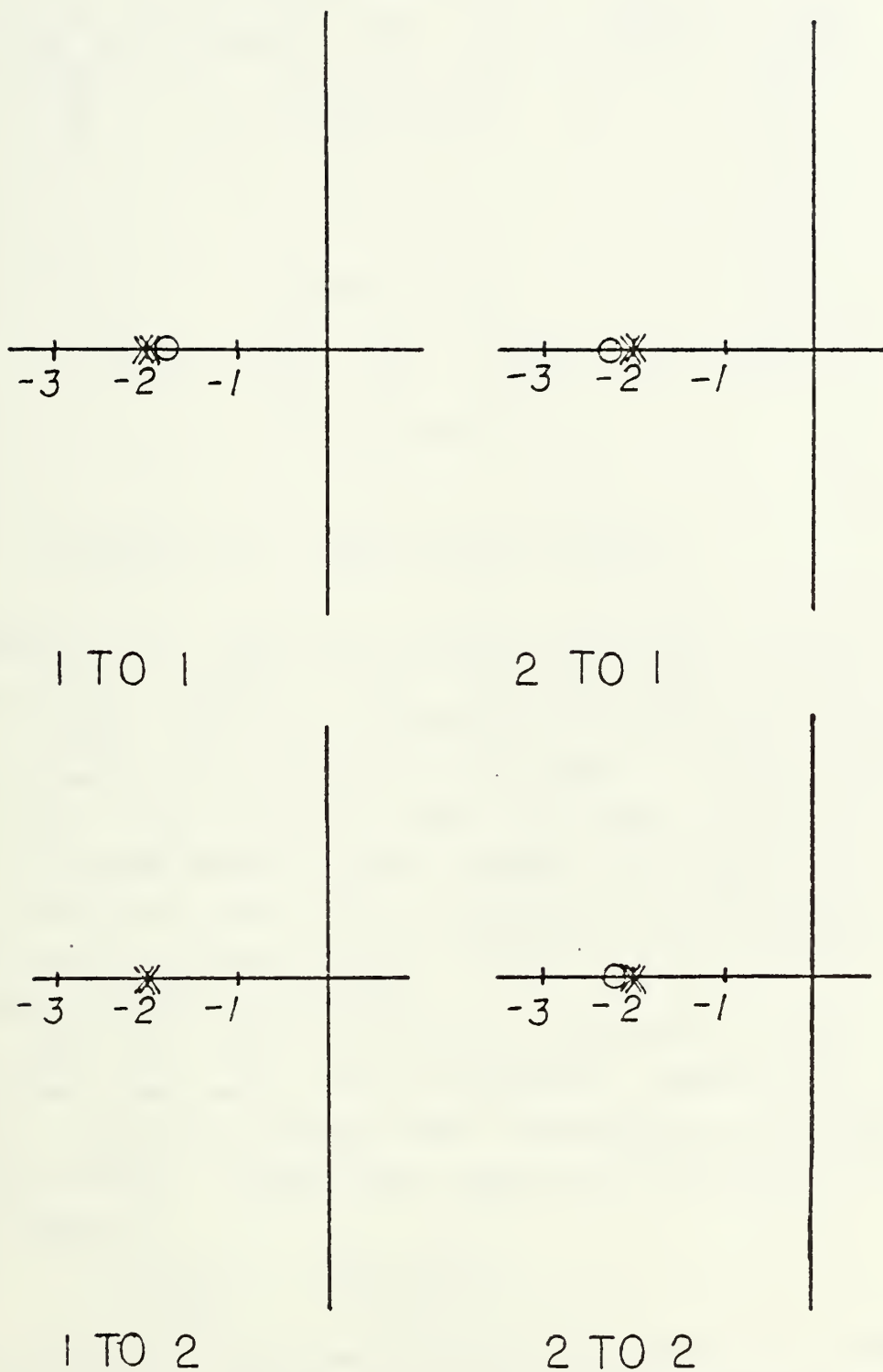


Figure 6.14 Pole-Zero Plot Simple Problem (4 Variables).

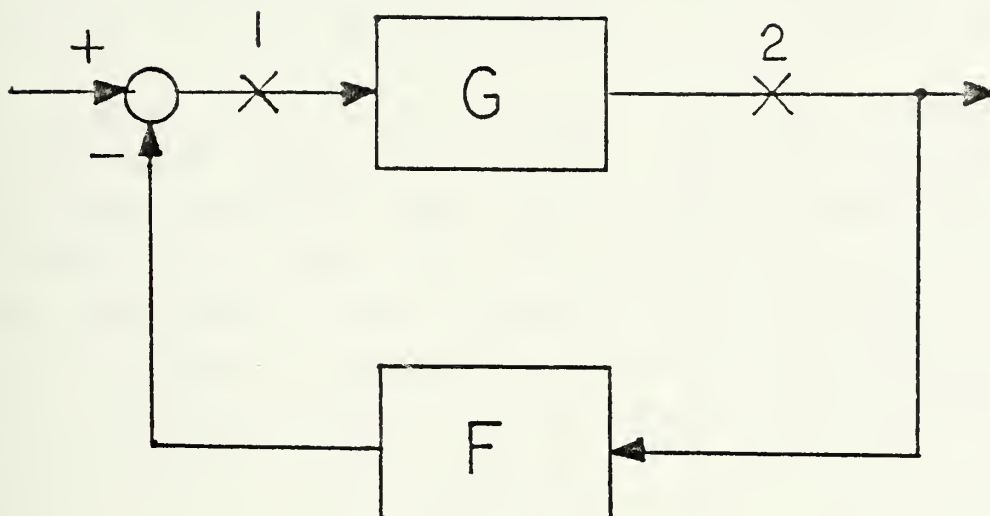


Figure 6.15 System Block Diagram.

pole placement and robustness routine a design was made in which the output singular values were specified instead of the input values. The pole placement and robustness routine produced a design with gains that provided a highly robust system on the output side with gain and phase margin of -6 db to and phase margin of ± 60 degrees. The design was not robust on the input side. Thus, designing for robustness at one point in the system does not necessarily give robustness at all points within the system.

One final case that should be discussed is that of setting both input and output robustness criteria at the same time. Excellent results were obtained for this case. The design routine placed the poles at $-1.97 \pm 0.009j$ with feedback gains of

$$\begin{array}{ll} 0.85793 & -45.91757 \\ 0.00425 & 0.87137 \end{array}$$

The input singular values were raised to a level of 0.74984 or -4.5 db to 12 db gain margin and ± 43 degrees phase

margin. The output values were raised to above 0.822 which corresponds to -5 to 15 db gain margin and ± 49 degrees phase margin.

To summarize, it can be stated that the robustness problem for this system exists in the upper cross-coupling channel (input two, output one). The lack of robustness can be discovered in two ways. The first method is to plot the open-loop Bode plots of each element of the transfer matrix and look for extremely high gains and bandwidths relative to the other transfer functions. The second method examines the singular values of the return difference matrix for magnitude. Low singular values correspond to low robustness. The pole placement and robustness design routine can increase robustness by modifying feedback gains to reduce the effect of cross-coupling within the system. Observing the gain modification made by the pole placement and robustness routine the critical channel within the system that affects the robustness may be determined from the Bode plots. The pole placement and robustness routine feedback gain changes also cause zero shifts during the robustness recovery. The gain on the open loop Bode plot for the affected cross-coupling channel is adjusted and the closed-loop zeros, as seen on the pole-zero diagram, are shifted. This zero shift is in a direction which will combine with system poles to smooth the frequency response diagram in the vicinity of the minimum singular value.

VII. A HELICOPTER STABILITY PROBLEM

This chapter will deal with a more practical application of the numerical optimization program. In this problem the combined pole placement, robustness design procedure will be applied to the linear lateral dynamic channels of a CH-47 helicopter. The model is a highly coupled two-input two-output system that has been studied for its basic robustness characteristics [Ref. 16]. The usual procedure for design of highly coupled systems is to obtain a diagonally dominant closed-loop system. This diagonal system will be stable but not robust to cross-feed parameters. Sandell, et al, produced three designs. Two of these designs, while meeting basic performance criteria, had poor robustness. The third design was a relatively good design. The numerical optimization technique developed in the thesis was applied to the two poor designs and shown to provide substantial improvement in robustness.

The systems were designed to satisfy specifications to step input response and stability margins as stated in military specifications. Specific design parameters for each of the three designs presented were not available. The designs were all considered to meet the performance specification criteria and stability margin requirements. It was shown that two of these designs were extremely sensitive to model errors. The classic Nyquist techniques did not predict this sensitivity. The singular value analysis did indicate sensitivity problems.

The model is that of a CH-47B helicopter lateral dynamic system in hover. The dynamic model of the system is

$$\dot{\underline{x}} = \underline{A} \underline{x} + \underline{B} \underline{u} \quad (7.1)$$

$$\underline{x} = (v, p, r, \varphi) \quad (7.2)$$

$$\underline{u} = (\delta_B, \delta_C) \quad (7.3)$$

where

$$\underline{A} = \begin{bmatrix} 2.27 & -1.42 & -0.15 & 31.99 \\ 0.01 & -0.7 & -0.07 & 0 \\ 0.04 & -0.05 & -0.5 & 0 \\ 0 & 1 & 0.11 & 0 \end{bmatrix}$$

and where

$$\underline{B} = \begin{bmatrix} 0.12 & 0.95 \\ 0.04 & -8.37 \\ .34 & 0.02 \\ 0 & 0 \end{bmatrix}$$

with full state available for feedback. Table 3 is a summary of parameters. The system is not open-loop stable.

Three control laws are formulated to satisfy the desired performance specifications. Equation 7.4 is the basic control law.

$$\underline{u} = -\underline{F}_i \underline{x} + h_i \varphi_c \quad (7.4)$$

$i=1, 2 \text{ or } 3$

where the following values of F and h were are:

$$\underline{F}_1 = \begin{bmatrix} -1.72 & -23.5 & 70.6 & 595. \\ 0.024 & -2.71 & 0.368 & -7.99 \end{bmatrix}$$

$$\underline{F}_2 = \begin{bmatrix} 0.198 & 154.0 & 18.3 & 142.0 \\ -0.01 & -1.592 & -0.189 & -1.47 \end{bmatrix}$$

$$\underline{F}_3 = \begin{bmatrix} 0 & 0 & 25.5 & 0 \\ 0 & -4 & 0 & -27 \end{bmatrix}$$

TABLE 2
CH-46 Helicopter Parameter Definitions

Variable	Units	Description
v	ft/sec	Vehicle body y-axis earth relative velocity component
p	rad/sec	Roll rate
r	rad/sec	Yaw rate
ϕ	rad	Roll attitude angle
δ_B	in	Yaw rate rotor deflection control
δ_C	in	Roll rate rotor deflection control

$$h_1 = \begin{bmatrix} 597.0 \\ -7.99 \end{bmatrix}$$

$$h_2 = \begin{bmatrix} 142.0 \\ -1.47 \end{bmatrix}$$

$$h_3 = \begin{bmatrix} 0 \\ -27 \end{bmatrix}$$

The ψ_c is a step input command that must track. Figure 7.1 is a diagram of the control structure for the feedback control laws. Figure 7.2 is a detailed layout of the system. All three of the designs have negative real eigenvalues and provide stable overdamped responses. All three of the above designs meet the desired design specifications.

The loop Bode plots, [Ref. 16], indicate all three designs to have good stability margins when considered a loop at a time. Since these designs were all full state IQ

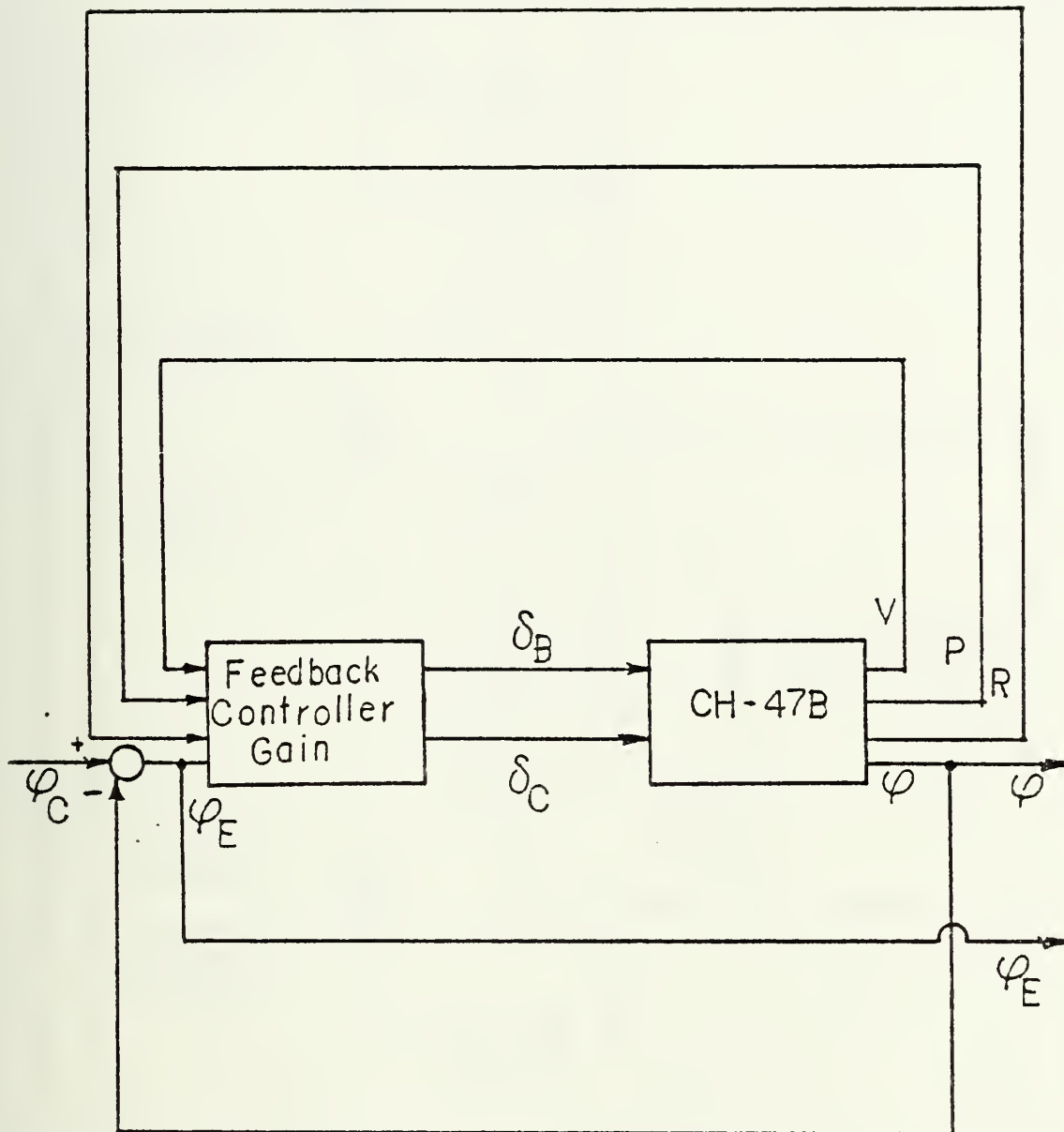


Figure 7.1 Feedback Control Structure.

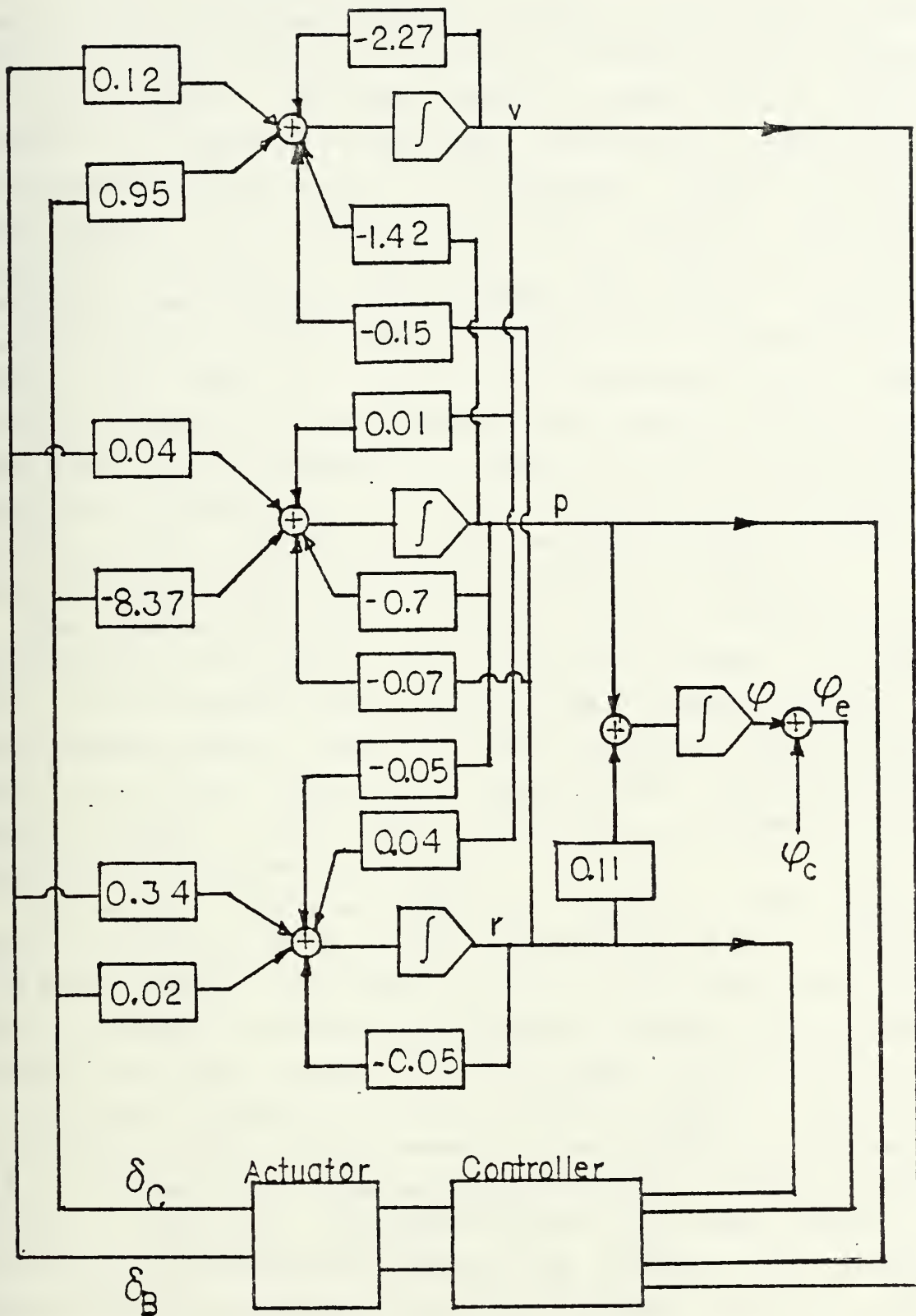


Figure 7.2 System Diagram.

designs with a diagonal control weighting matrix they should possess at least -6 db to infinite gain margin and 60 degrees of phase margin. When the singular values of the three designs are computed a robustness problem is indicated by low singular values of two of the designs. Figure 7.3 presents singular value plots of all three designs. This plot shows that designs 1 and 2 both have very low minimum singular values for the return difference matrix. Design 1 goes as low as -20 db near 10 rad/sec in frequency while design 2 is down to -34 db at frequencies up to 1000 rad/sec. Design 3 is a good design with singular values that remain above one throughout the frequency range of interest. Using the universal gain and phase diagram as discussed in Chapter 6 this equates to a gain margin of about -6 db to infinity and a phase margin of 60 degrees as expected.

The system stability in design 1 may be affected by perturbations occurring in the actuators as shown in figure 7.4. If the output axis coupling from δ_C spills into δ_B in the frequency range from 0.5 to 50.0 rad/sec with a magnitude of 0.12 and a phase of 60 degrees then the system can become unstable. This could be caused by nonlinear terms, worn parts, or system saturation.

The second design as shown in figure 7.5 may have a stability derivative variation between δ_B and p. If this derivative varies from about 0.04 to -0.96 the system can become unstable. The center of gravity location, trim of the aircraft and rotor coupling can all affect this derivative.

The pole placement and robustness design technique developed for this thesis research was applied to designs 1 and 2 to obtain an improved robustness for these designs. Design 1 will be considered first. In this design a cross-feed perturbation through the actuator can produce instability as indicated by the low singular values at lower frequencies. To study this problem with the pole placement

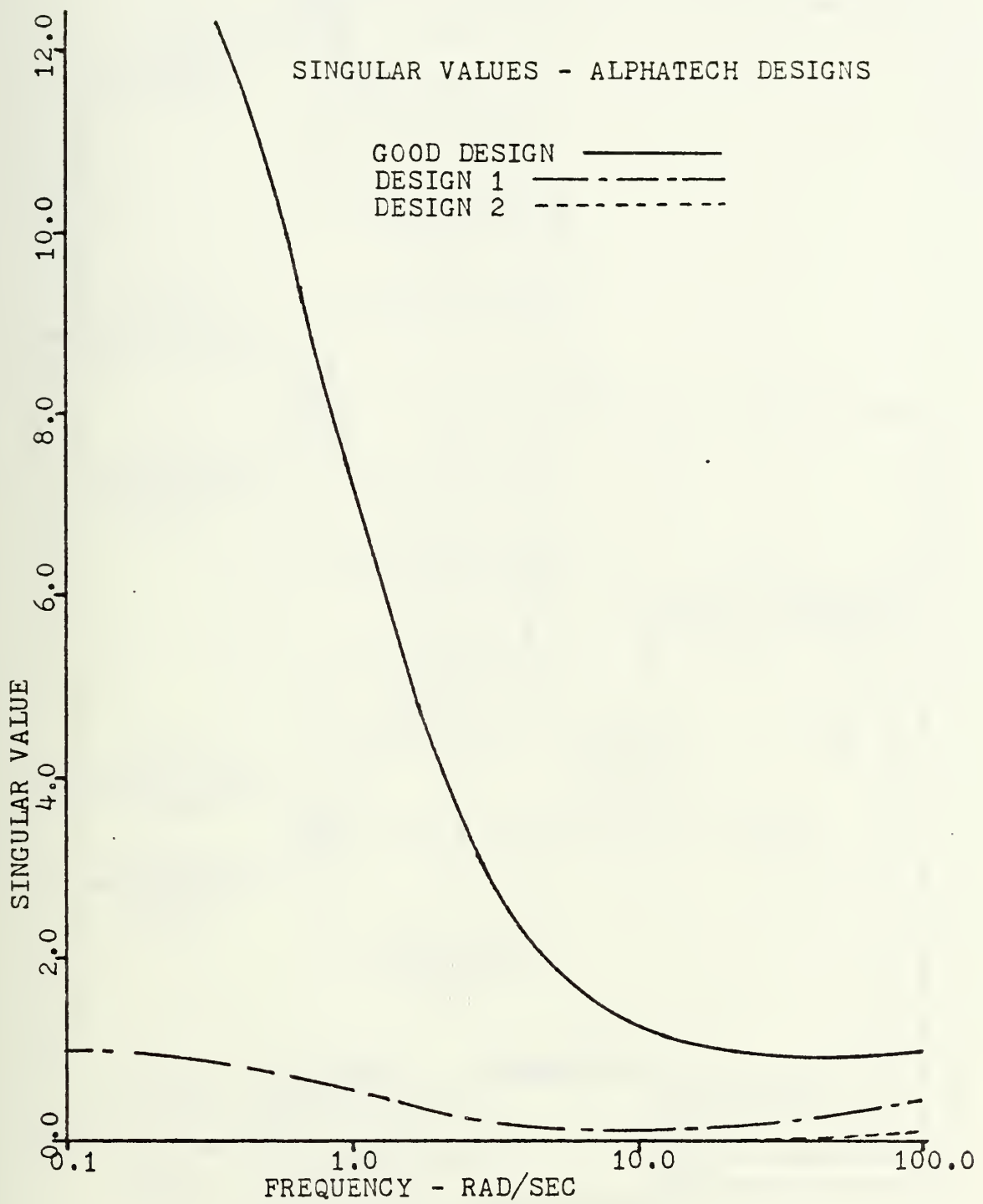


Figure 7.3 Alphatech Design Singular Value Plots.

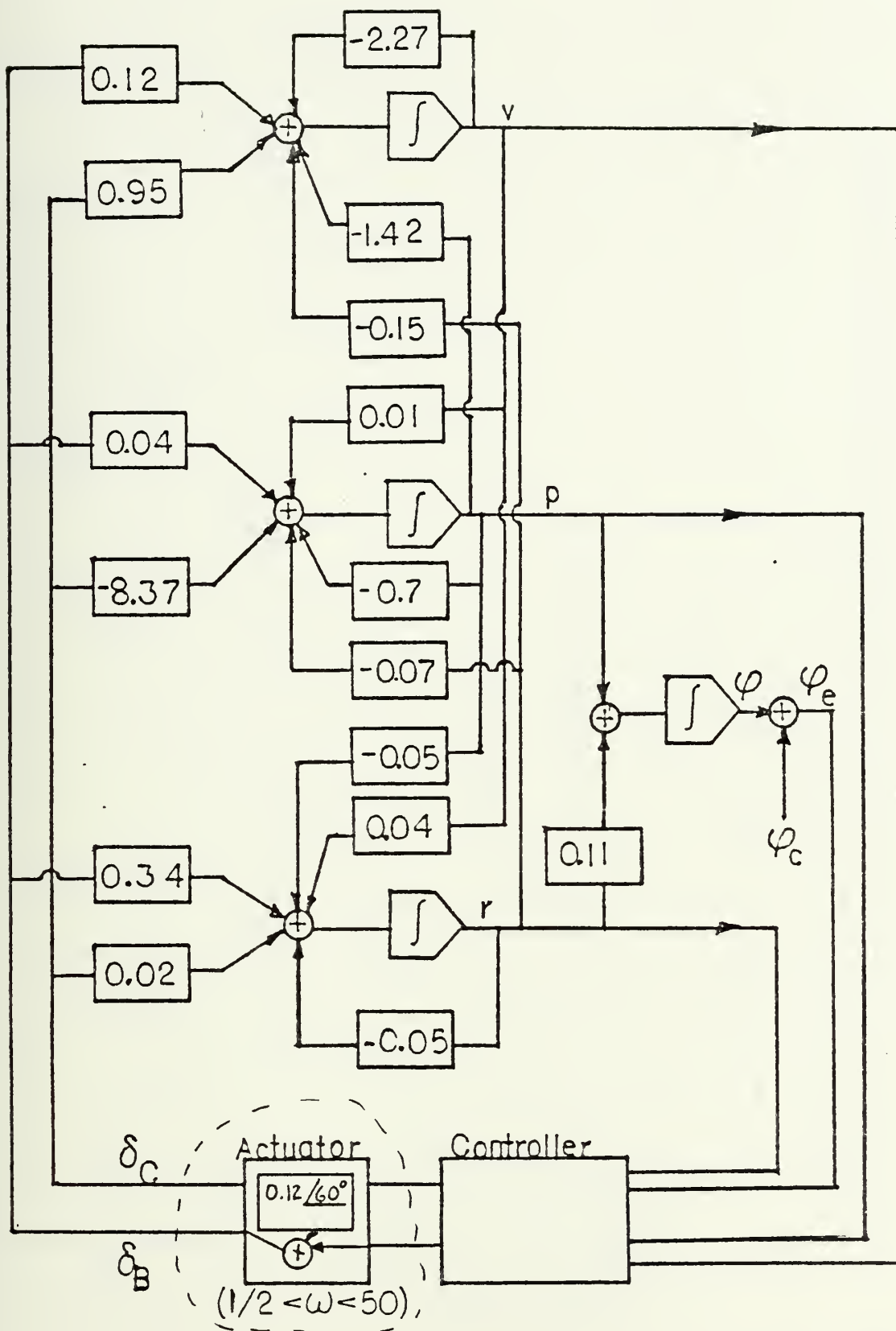


Figure 7.4 Design One Perturbation Input.

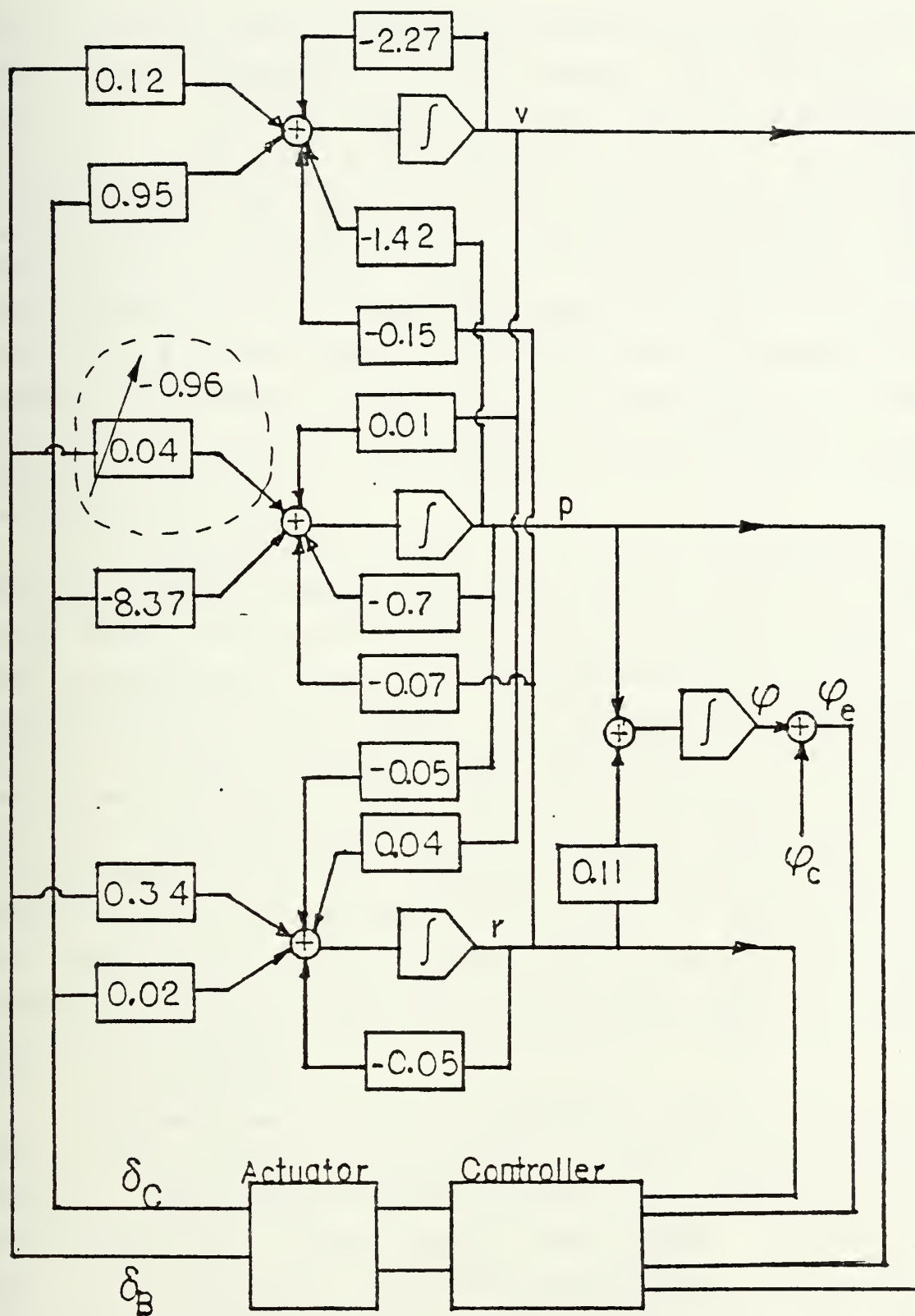


Figure 7.5 Design Two Perturbation Input.

and robustness design method it is assumed that the eigenvalues (pole locations) of the system as developed in [Ref. 16]. are the required poles for the performance criteria. Once the pole locations are set, robustness criteria must be selected. From the universal gain and phase margin curve discussed earlier two choices of singular value levels were made for this problem. The first singular value level chosen was 0.6. This corresponds to a gain margin of -4.0 db to 8 db and a phase margin of about 35 degrees. The second value chosen was 1.0. With corresponding gain margin of -6 db to ∞ and a phase margin of 60 degrees which are the characteristics of a LQ regulator design with diagonal weighting matrices.

For singular value level 0.6 the pole placement and robustness design routine places the poles as shown in table 2. The slight differences in these pole locations appear to have insignificant effect on the performance as shown in the response curves of the system. The feedback gain adjustment moves the minimum singular value from about 0.11 with very poor phase and gain margins to a level of 0.66. This is above the desired values of gain and phase. The improvement in robustness came from modifying the feedback gains in channel δ_B . By greatly reducing the gains in channel δ_B the optimizer minimizes the influence of the cross-coupling from channel δ_C . In this way a much larger spill over of channel δ_C may be tolerated through the actuator before the system will become unstable.

The primary mechanism of robustness improvement in this problem was a reduction in the gain levels of the affected channel. The feedback gains presented in table 4 show the modification of these gains from those utilized in [Ref. 16]. While all the gains are modified the f_{14} gain undergoes a significantly larger change than the other design 1 case 1 feedback gains. Looking at the open-loop

TABLE 3
Design One Pole Placement

Pole	Desired Location	Actual Location
1	-24.7977	-24.7893
2	-11.3635	-12.0083
3	-10.3288	-10.7752
4	- 2.1005	- 2.1181

transfer functions of this optimized problem and comparing them with a non-optimized set of transfer functions for δ_B to as shown in figure 7.6 and 7.7 show very little change in the system gain, both are around 42.0 db, and a bandwidth of about 20 rad/sec. There is a significant change in the phase diagram which is caused by the zero location shift.

The transfer function for δ_C to δ_C depicted a gain increase of about 3 db for the optimized design while the phase remained similar for both transfer functions. In transfer function δ_C to δ_B some important aspects of the problem are observed. The Bode diagram of the open-loop transfer function of δ_C to δ_B clearly indicates the cross-coupling problems and the pole placement and robustness design routine's mechanism of optimizing the system gains to increase robustness. The gain is reduced from 93 db to about 78 db and the bandwidth is reduced from above

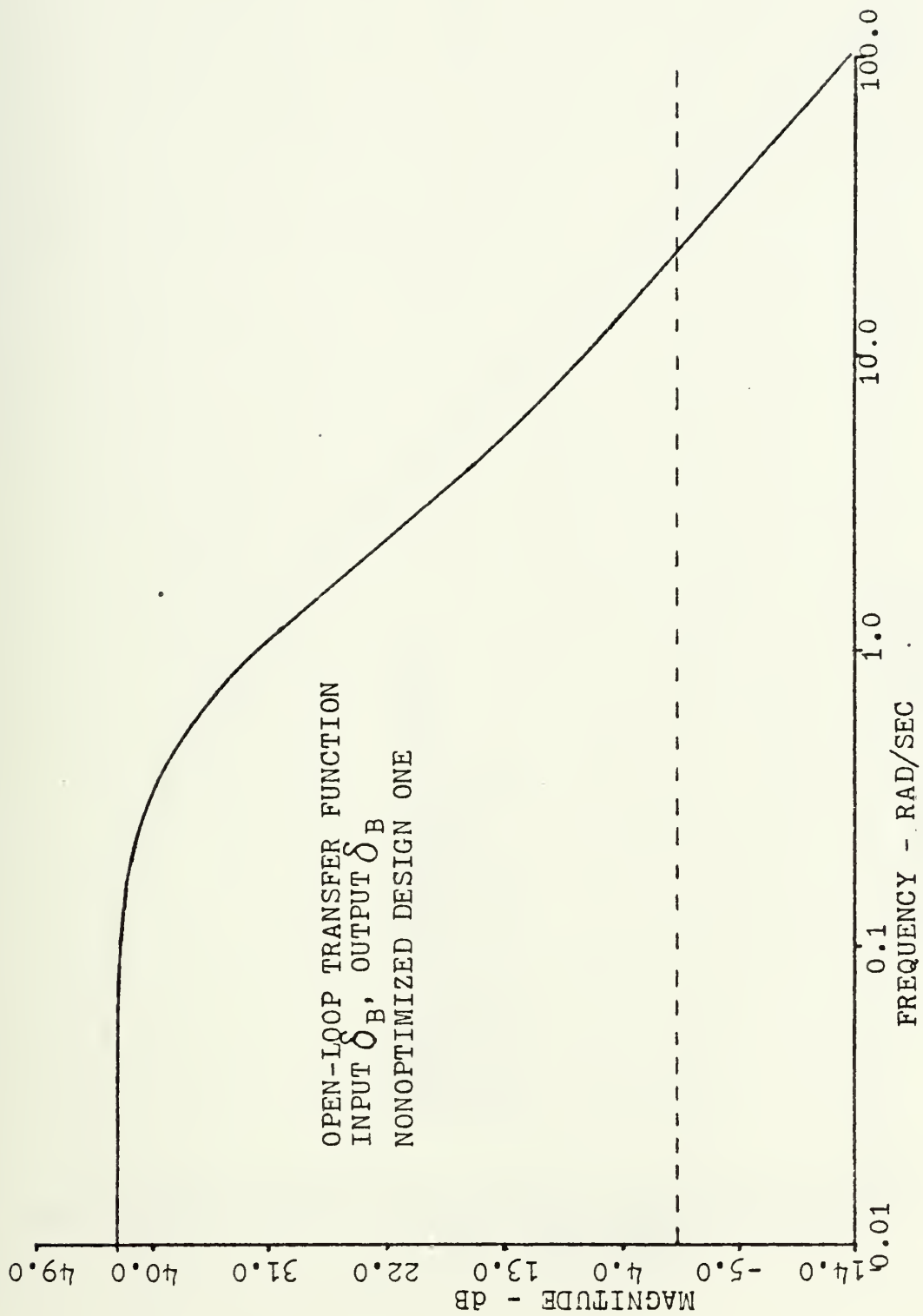


Figure 7.6 Transfer Function $\delta_B - \delta_B$ Design One, Nonoptimized.

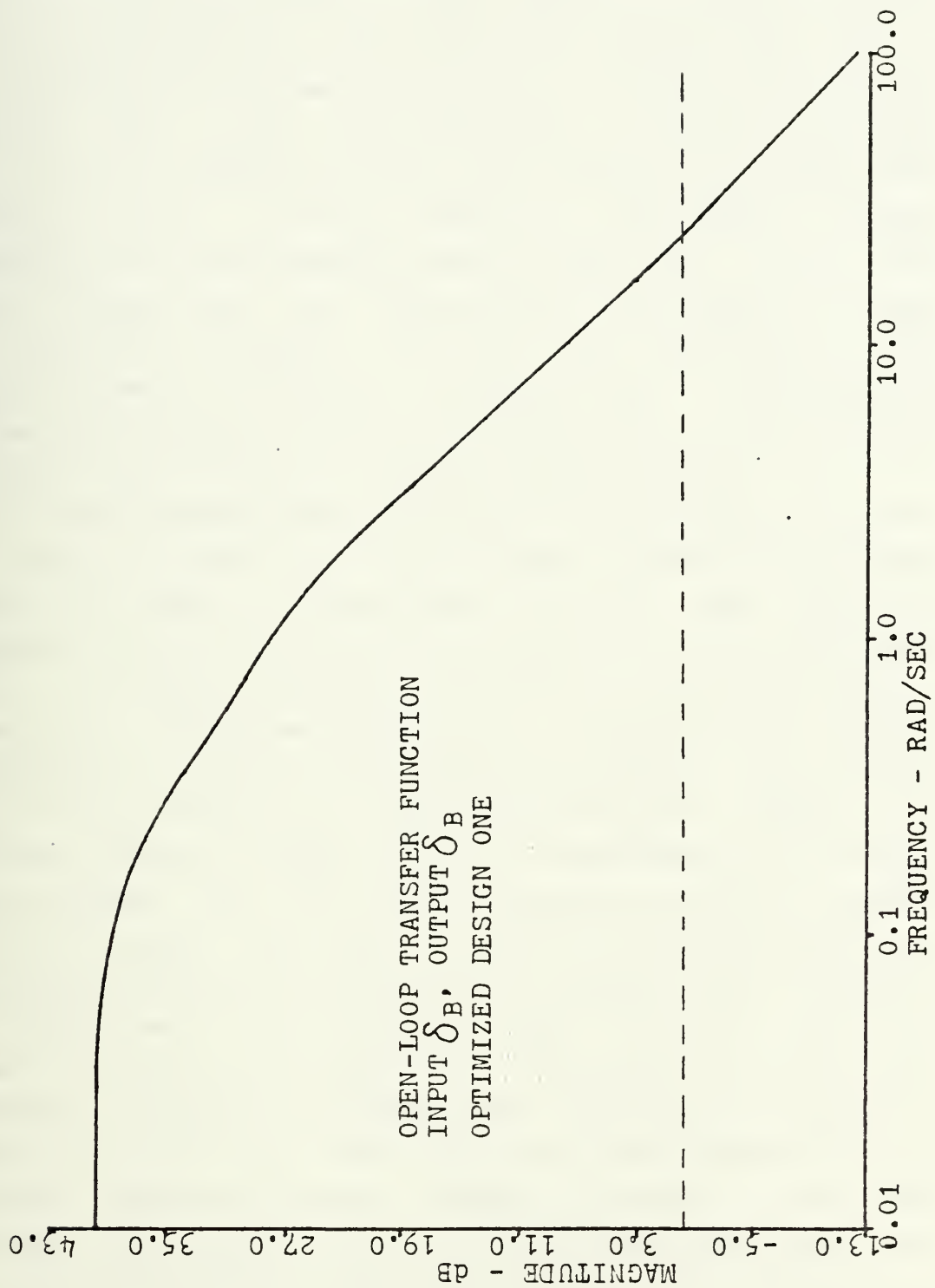


Figure 7.7 Transfer Function δ_B - δ_B Design One , Optimized.

100 rad/sec to about 30 rad/sec. The reduction of bandwidth and gain in the loop δ_c to δ_B yields an increased tolerance to perturbation. The transfer function for δ_c to δ_B is shown in figures 7.8 and 7.9. In the transfer functions for δ_B to δ_c the bandwidth is slightly increased from 0.6 to 0.8 rad/sec and the gain actually increased from 9 db to 14 db. The big change in the overall system, however, is in the transfer function from δ_c to δ_B . This is the channel that the destabilizing perturbation enters and by greatly reducing the gain and bandwidth in this channel through a change in feedback gain, the optimizer routine has brought the entire system gains to more balanced conditions and recovered a highly robust design.

The gain changes associated with the robustness improvement cause the zeros of the various closed-loop pole-zero diagram to move. A comparison of the eight pole-zero diagrams is shown in figures 7.10 to 7.13. The significant feature of these pole-zero diagrams is the shift of the zeros of the optimized design in a direction that attempts to equalize or balance the frequency response for frequencies in the vicinity of the minimum singular values. The pole-zero diagram of δ_B to v will be discussed as an example of this effect. In 7.10 the nonoptimized zeros are located about $-2 \pm 4.5j$ and -19.8 . When the pole placement and robustness routine has completed the feedback gain modification these zeros have shifted near -11 and $-6 \pm 3j$. The effect of these zero shifts is to combine with the pole locations to equalize the frequency response as depicted in figures 7.14 and 7.15. Zero shifts for the remainder of the transfer functions provide similar results in the other channels. By moving toward the frequencies associated with the minimum singular values the zeros have balanced the overall frequency response of the system in each channel. While the channel gain modification is the primary mechanism

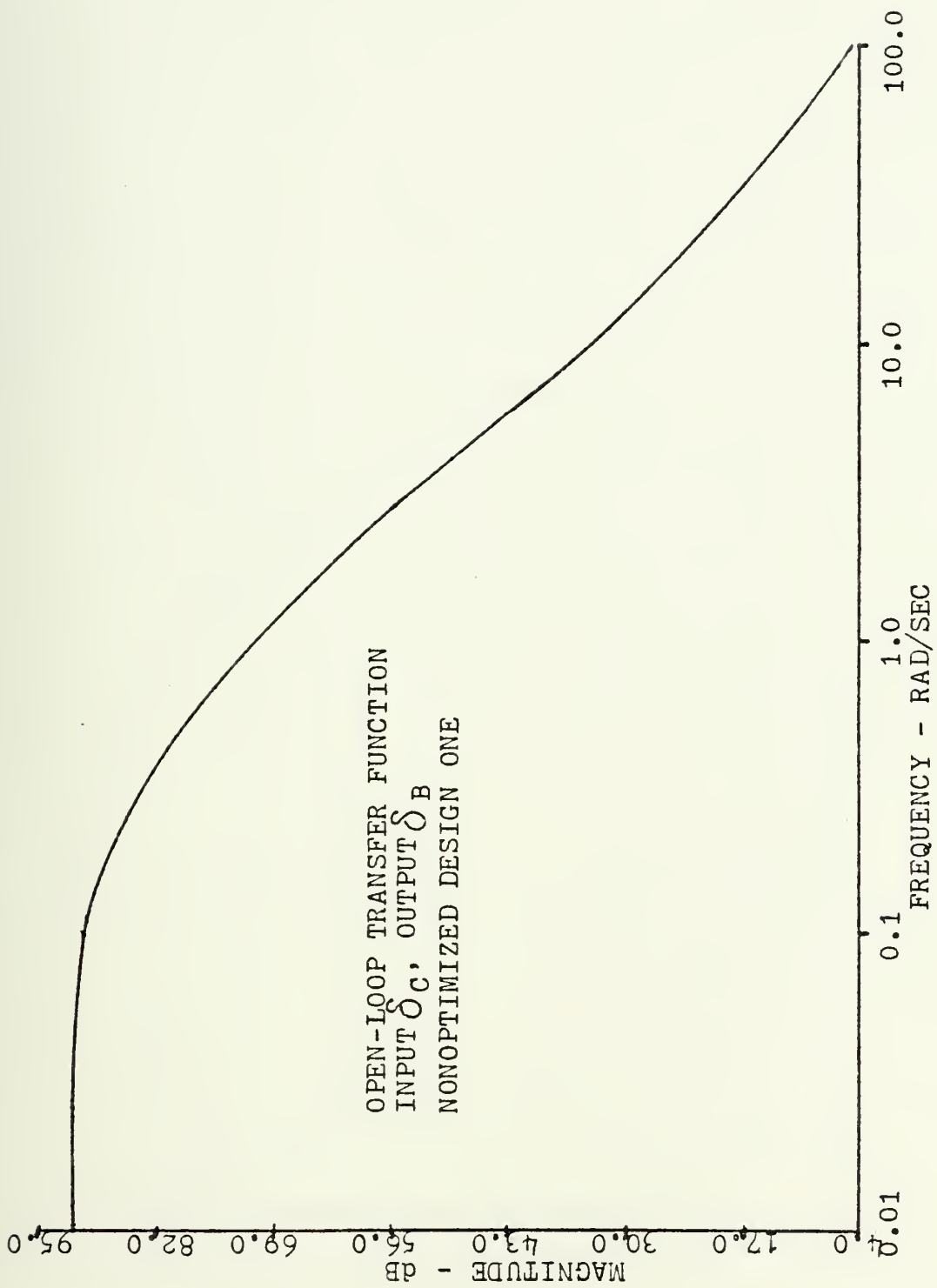


Figure 7.8 Transfer Function δ_C - δ_B Design One, Nonoptimized.

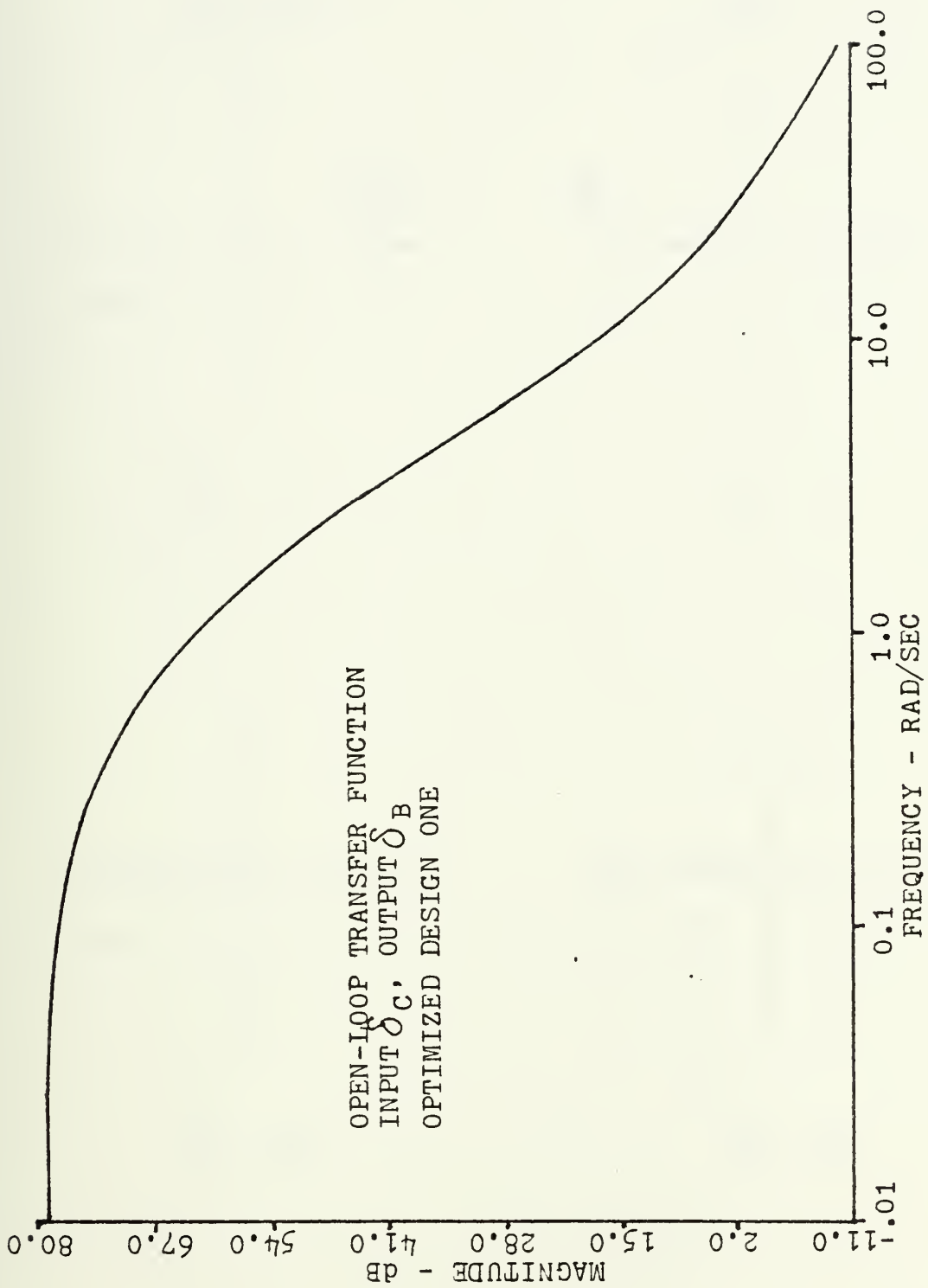
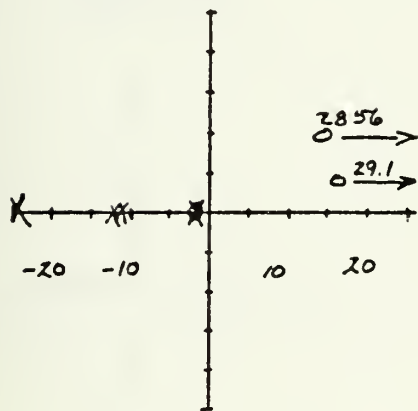


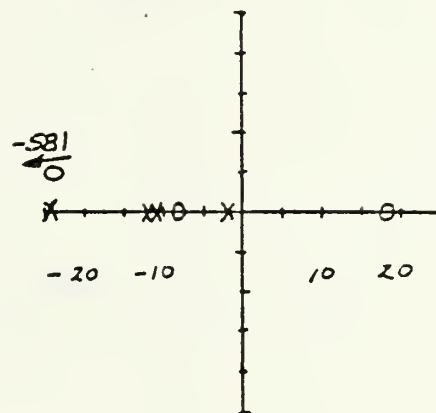
Figure 7.9 Transfer Function $\delta_C - \delta_B$ Design One, Optimized.

NON OPT

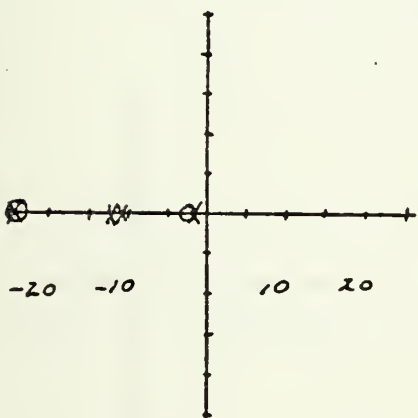


$\delta_C - r$

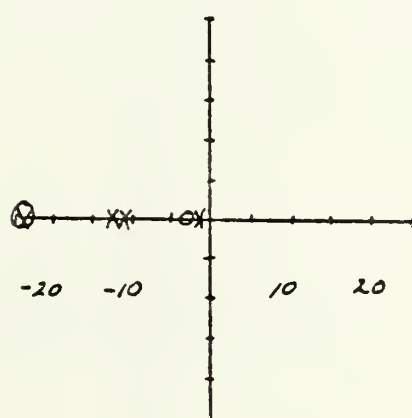
OPT



$\delta_C - r$



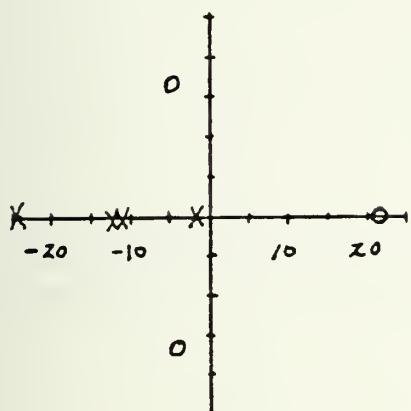
$\delta_C - \varphi$



$\delta_C - \varphi$

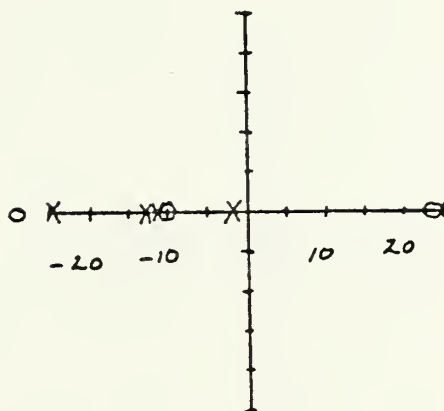
Figure 7.10 Pole-Zero Plots for Design 1.

NON OPT

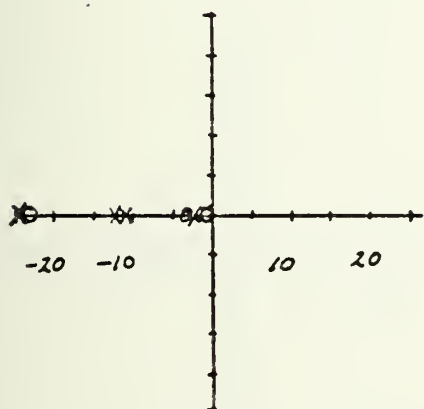


$\delta_C - V$

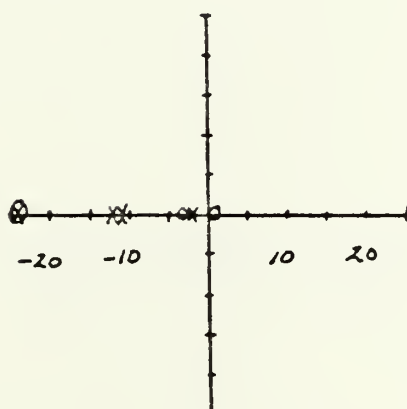
OPT



$\delta_C - V$



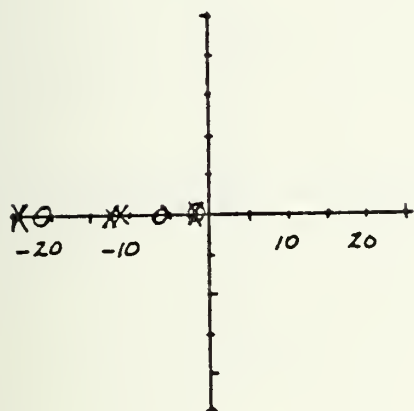
$\delta_C - P$



$\delta_C - P$

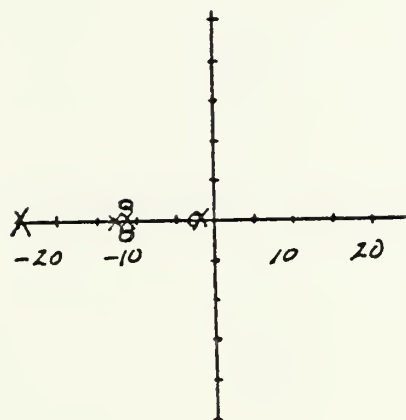
Figure 7.11 Pole-Zero Plots for Design 1 (cont.).

NON OPT

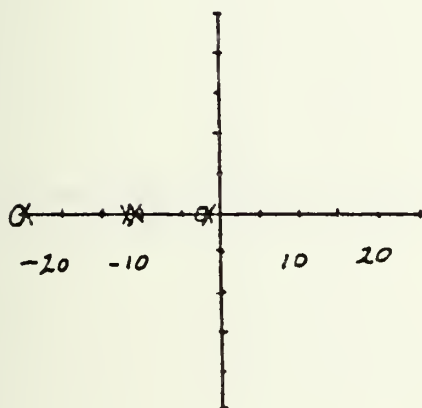


$\delta_B - r$

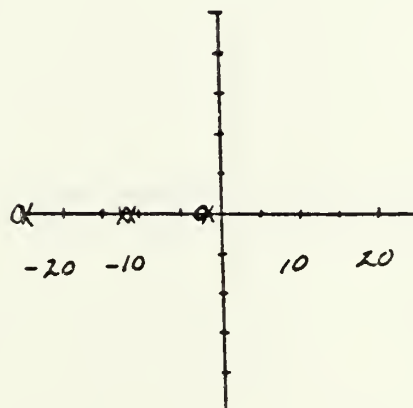
OPT



$\delta_B - r$



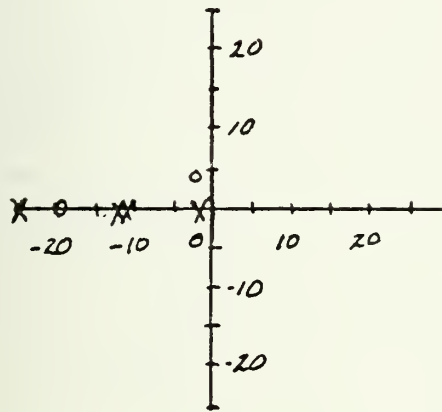
$\delta_B - \varphi$



$\delta_B - \varphi$

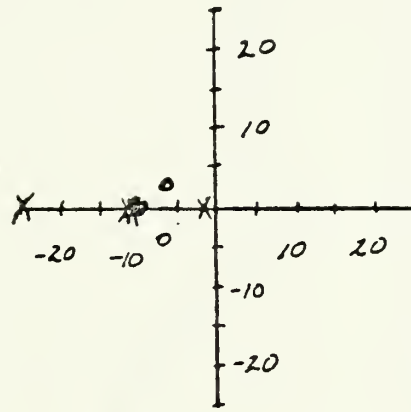
Figure 7.12 Pole-Zero Plots for Design 1 (cont.).

NON OPT

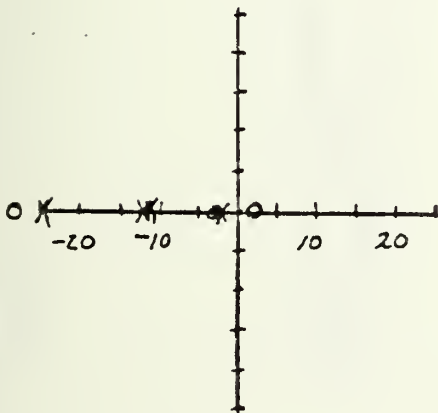


$\delta_B - V$

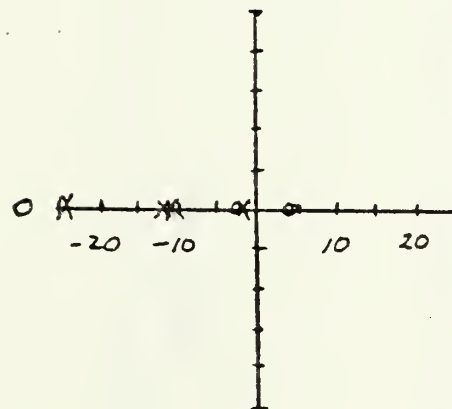
OPT



$\delta_B - V$



$\delta_B - P$



$\delta_B - P$

Figure 7.13 Pole-Zero Plots for Design 1 (cont.).

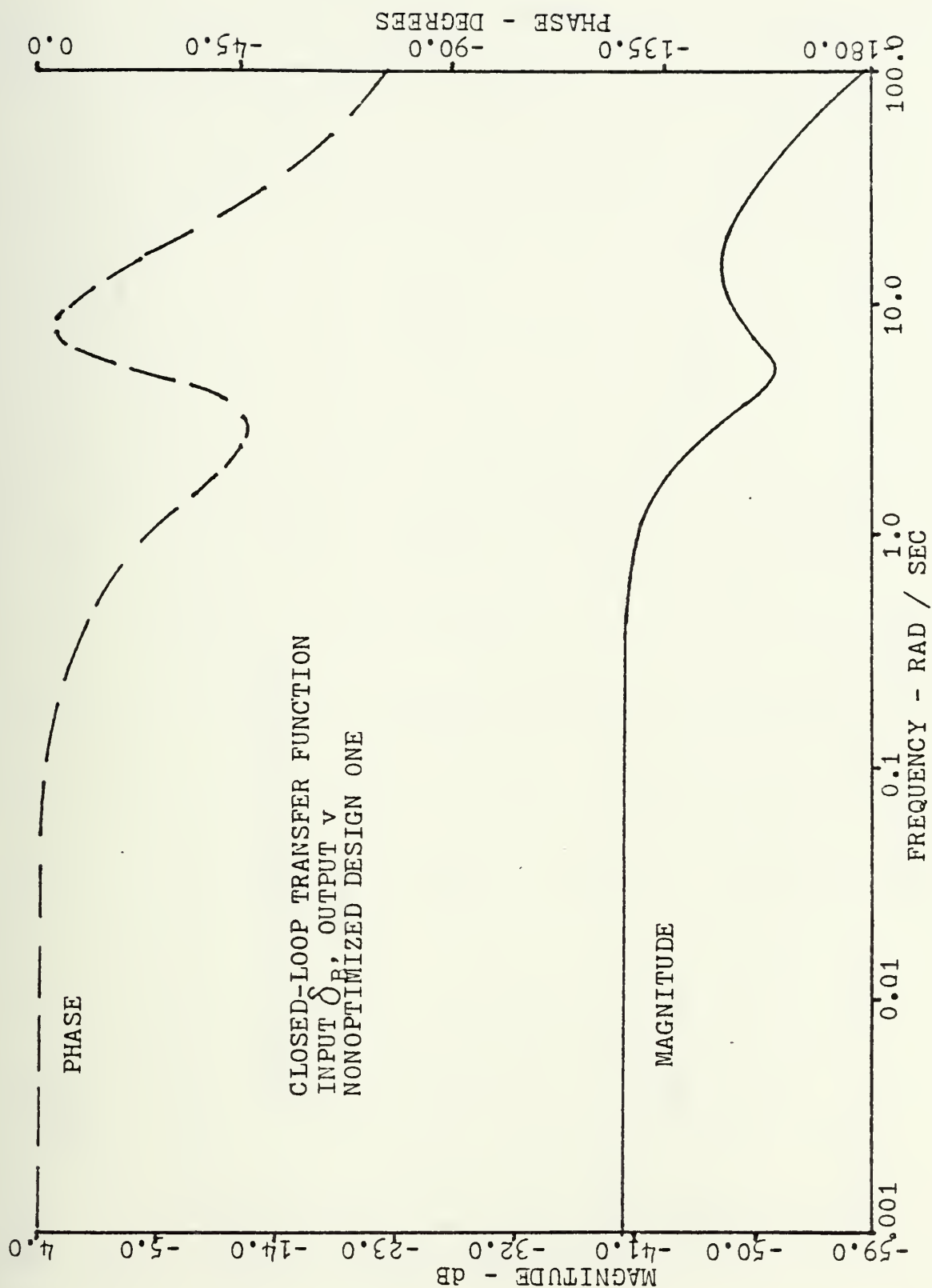


Figure 7.14 The δ_B to v Frequency Response, Nonopt..

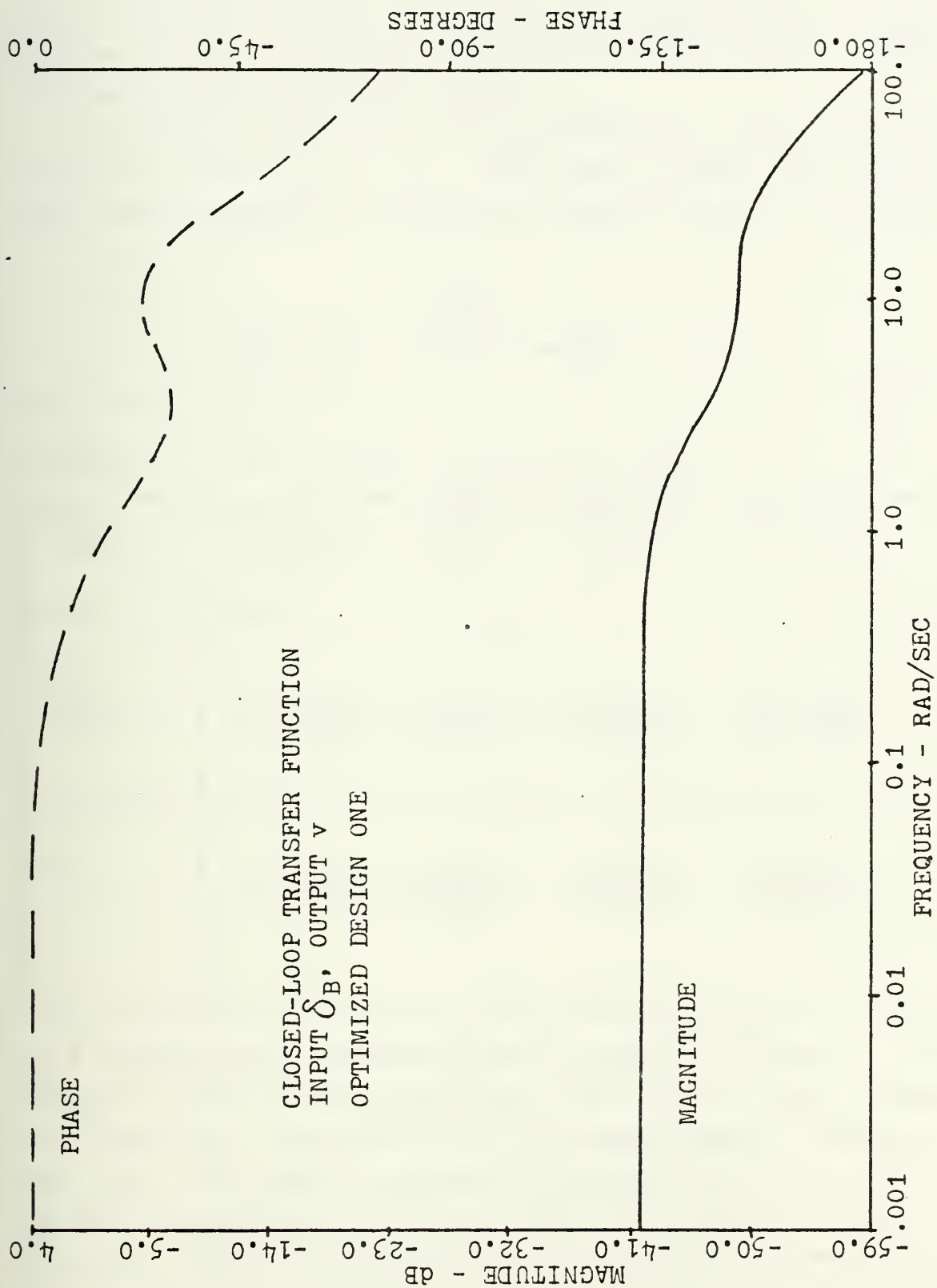


Figure 7.15 The δ_B to v Frequency Response, Optimized.

for robustness recovery the zero shift associated with the feedback gain changes is directly related to the overall frequency response of the system.

In design 1 case 2 the minimum singular value level was increased from 0.6 to 1.0. The pole placement and robustness design procedure also gave a better design for case 2

TABLE 4
Helicopter Problem Feedback Gains

DESIGN	GAIN VALUES $\begin{bmatrix} f_{11} & f_{12} & f_{13} & f_{14} \\ f_{21} & f_{22} & f_{23} & f_{24} \end{bmatrix}$			
One Case 1	-14.77726 -0.00567	2.15858 -2.55646	77.96629 0.39039	-32.91595 -15.04805
One Case 2	-6.36249 0.00813	-1.53746 -2.60073	72.85013 0.65395	74.20387 -13.47128
Two	-0.96660 0.00916	-4.54597 -3.03069	0.75436 -0.36391	11.96808 -0.34991

by modifying the feedback gains as shown in table 4. The change in the transfer functions for δ_c to δ_B (figure 7.16 and 7.17) is similar to the change seen in figures 7.8 and 7.9. The gain was reduced from above 90 db to about 75 db and the bandwidth cut from above 100 rad/sec to about 25 rad/sec. The added requirement of increased robustness did not result in a significant change in performance. [Ref. 16].

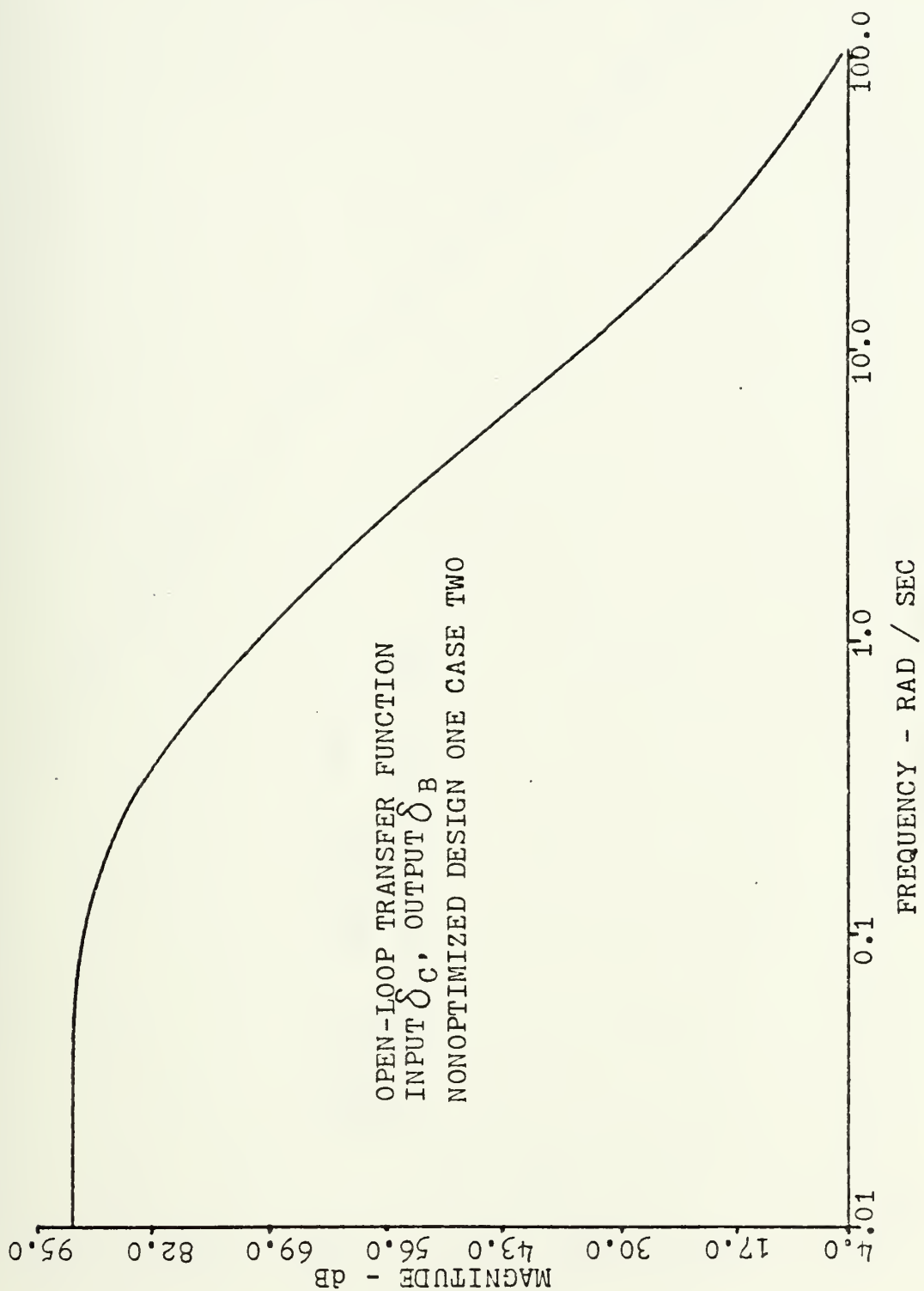


Figure 7.16 Transfer Function $\delta_C - \delta_B$ Design 1 Case2, Nonopt..

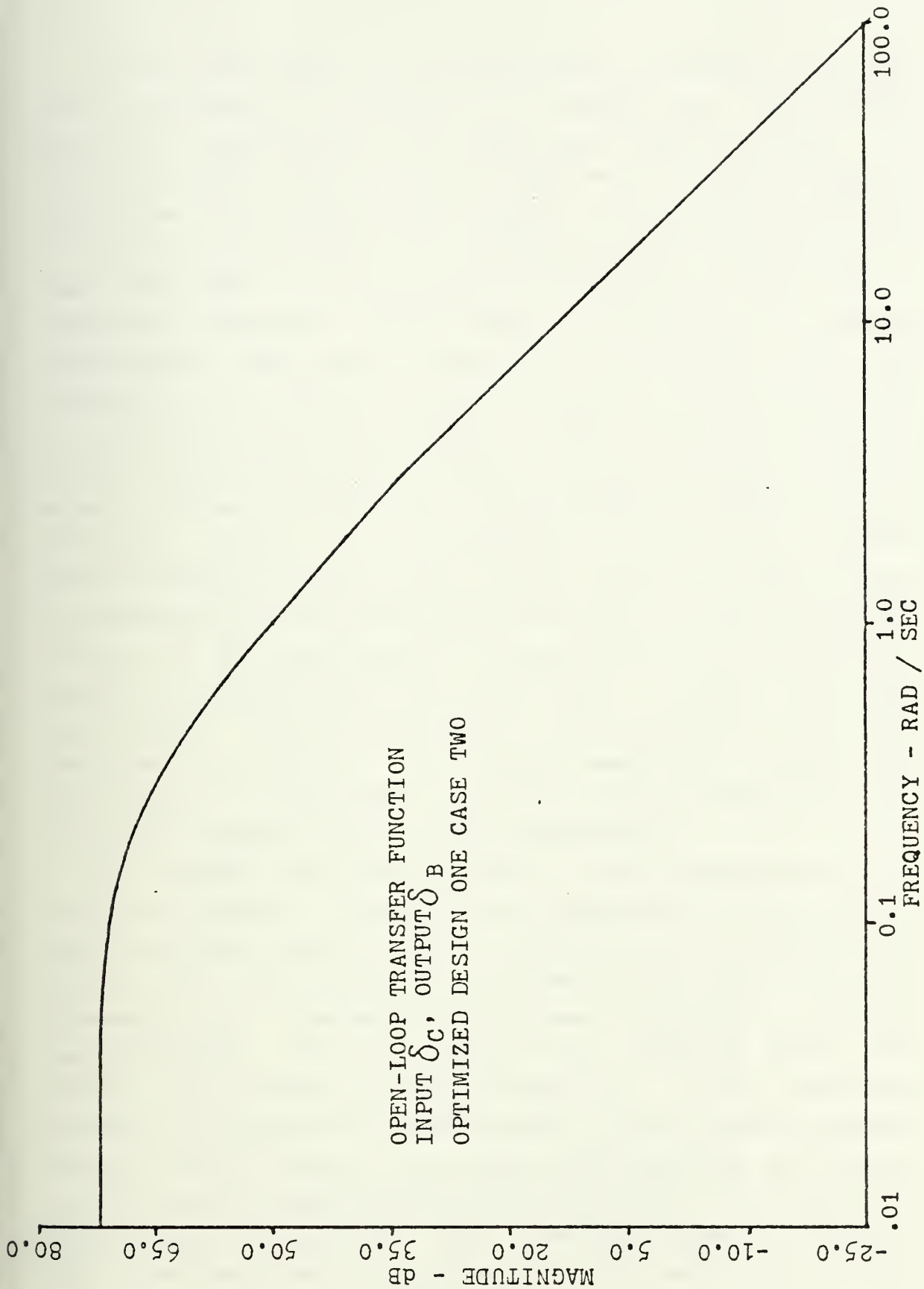


Figure 7.17 Transfer Function $\delta_C - \delta_B$ Design 1 Case 2, Opt.

Design two which is the design that is subject to parameter uncertainty in the δ_B to p channel was also studied using the pole placement and robustness design routine. In this design the singular values were very low until frequencies above 100 rad/sec were reached. To improve the design it was assumed that the pole locations that corresponded to [Ref. 16] were the required pole locations and that a robustness singular value level of 0.6 would provide adequate gain and phase margin for the design. The pole placement and robustness routine adjusted the gains in this problem until a minimum singular value of 0.6 was obtained. During this adjustment the gains in channel δ_B were considerably reduced to offset the cross-coupling between the two channels. Again, in this design as before the channel cross-coupling between δ_C and δ_B shows a marked change in bandwidth and gain from above 100 to 35 rad/sec and about 78 db to 22 db respectively. Figures 7.18 and 7.19 show the transfer function plots for this term. Figure 7.20 shows the singular value improvement. In figure 7.21 the time response is plotted. As can be seen in the plot the improvement in robustness for this problem results in very sluggish response and degraded performance.

To summarize for this problem, a given performance level has been chosen in terms of pole locations. The level of robustness has been set for a desired gain and phase margin based on the universal gain and phase margin curve. The pole placement and robustness routine has been able to improve the robustness level by changing the feedback gains that affect the channel δ_C cross-coupling. This robustness recovery is affected by modification of the system feedback gains in such a manner that cross coupling gains are reduced so that small cross-coupling perturbations do not drive the system into instability. The open-loop transfer function plots have been used to indicate how this mechanism operates

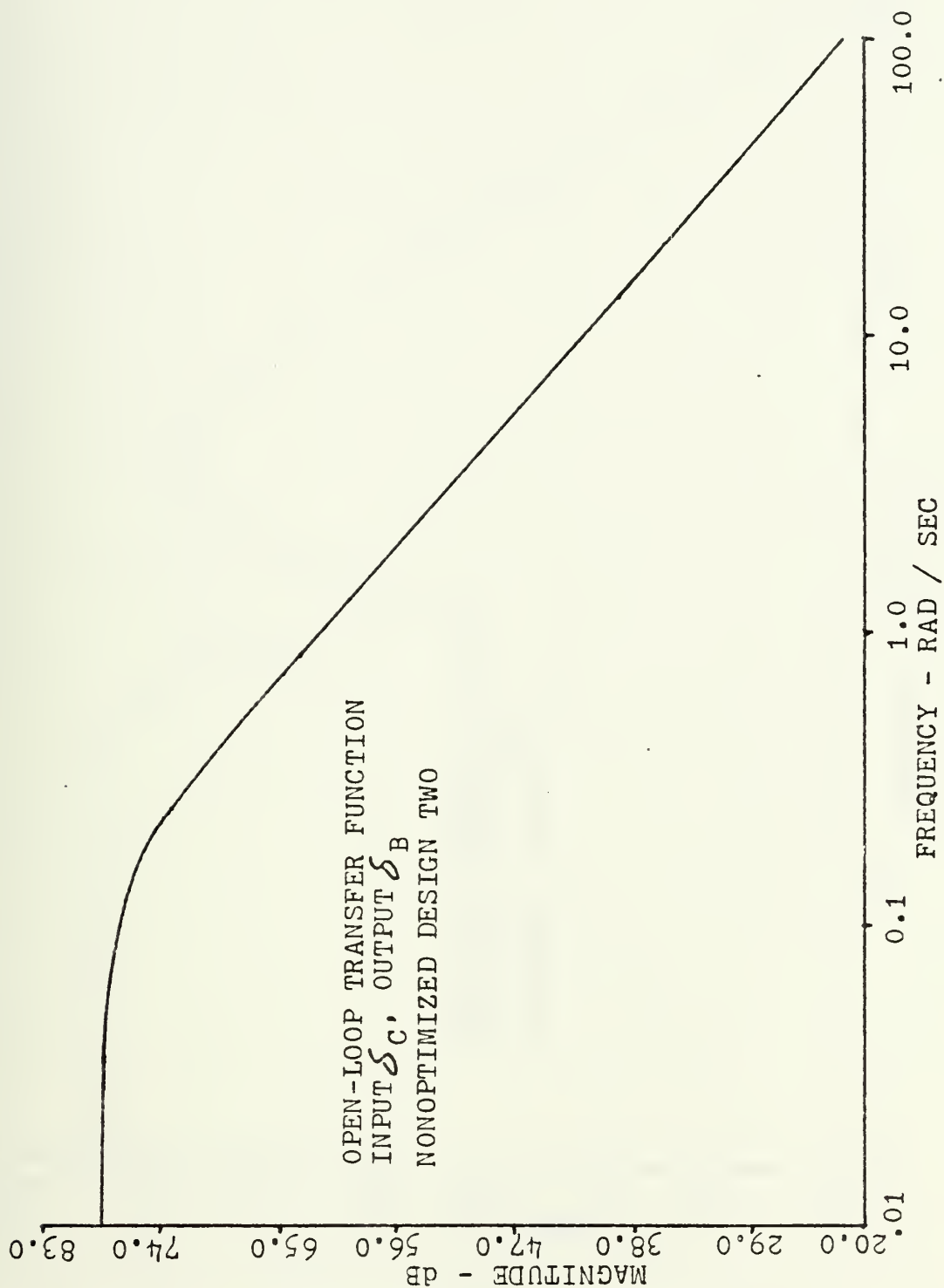


Figure 7.18 Transfer Function $\delta_C - \delta_B$ Design Two, Nonopt..

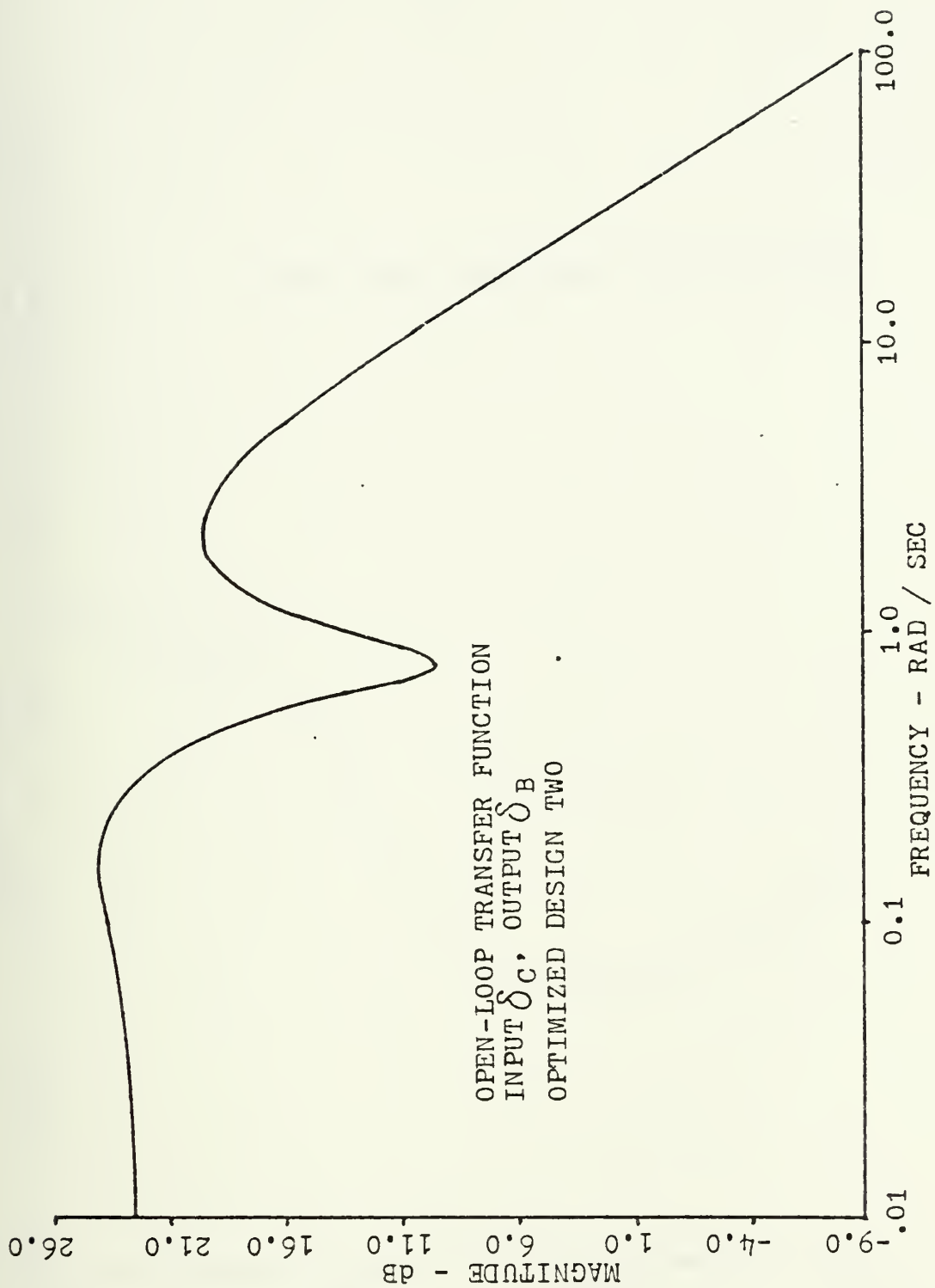


Figure 7.19 Transfer Function $\delta_C - \delta_B$ Design Two, Optimized.

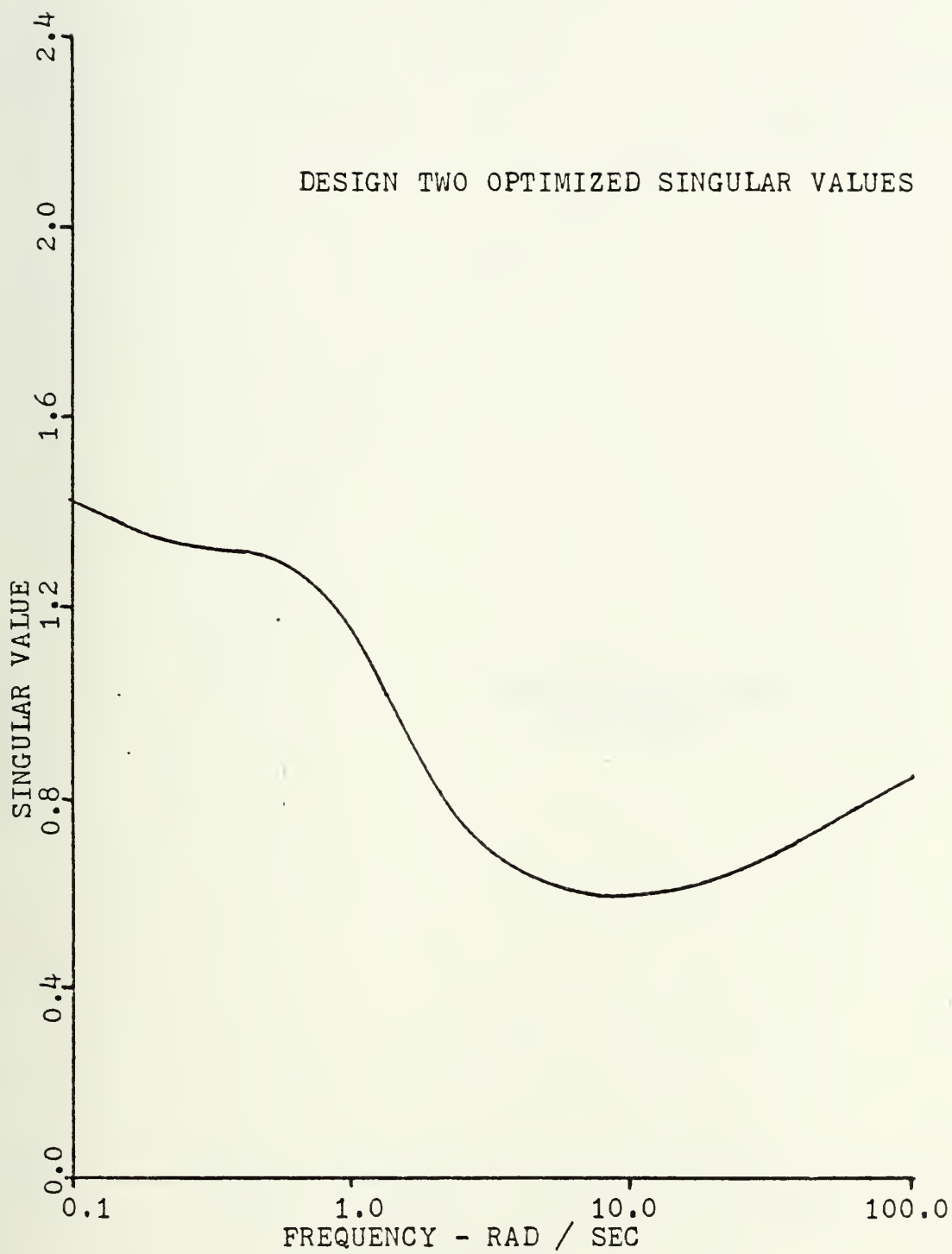


Figure 7.20 Singular Value Plot Design Two.

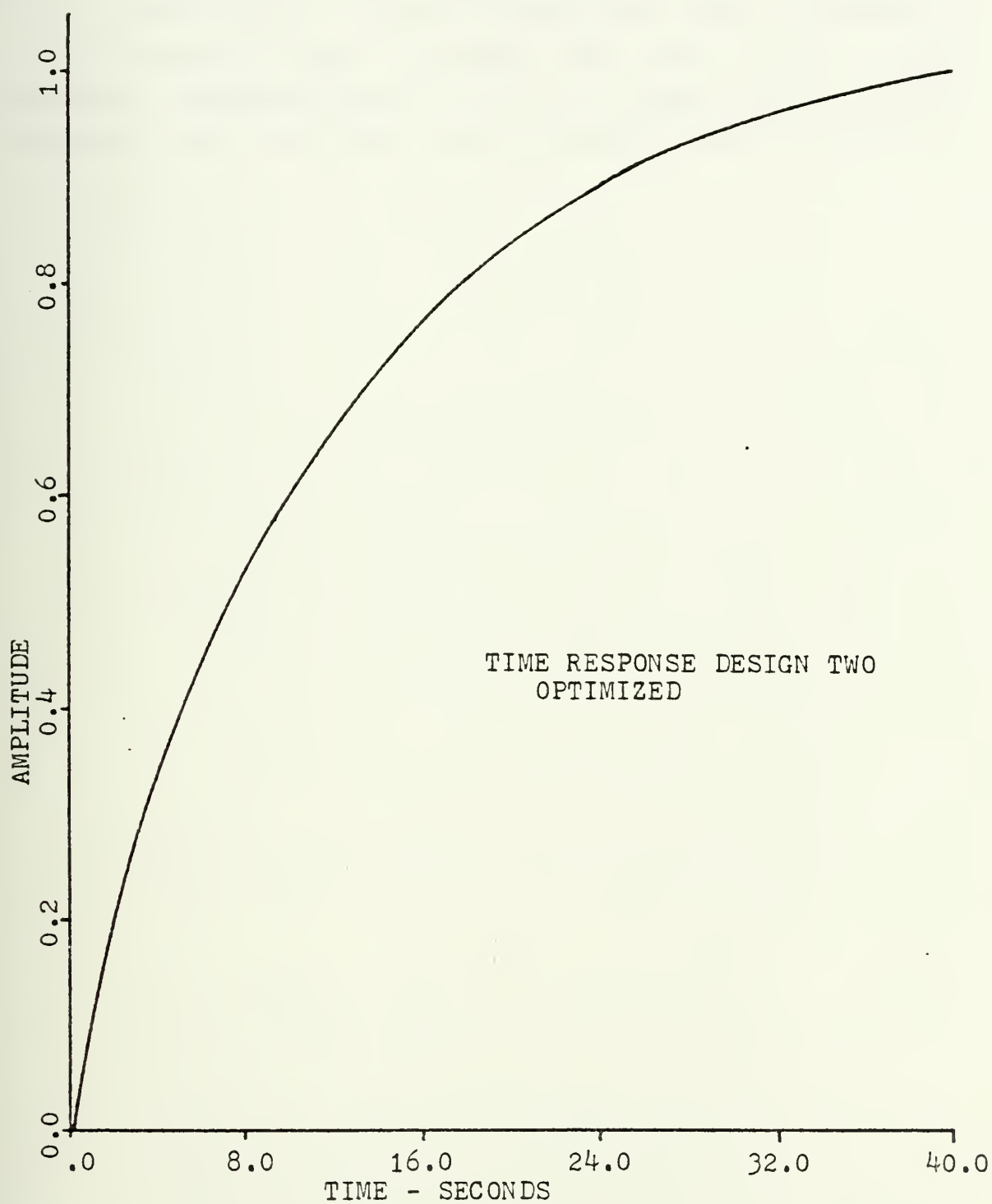


Figure 7.21 Time Response Design Two.

and have been shown to be an alternative indicator of channels that may be affected by cross-feed perturbations. The pole-zero diagrams of the closed-loop transfer functions of the transfer matrix further indicate that zero movement is in a direction that equalizes the gain level of the frequency response curves in the vicinity of the lowest singular values providing a more balanced system response.

VIII. SIMPLE OBSERVER

The pole placement and robustness design procedure can also be used for robustness recovery in observer design.

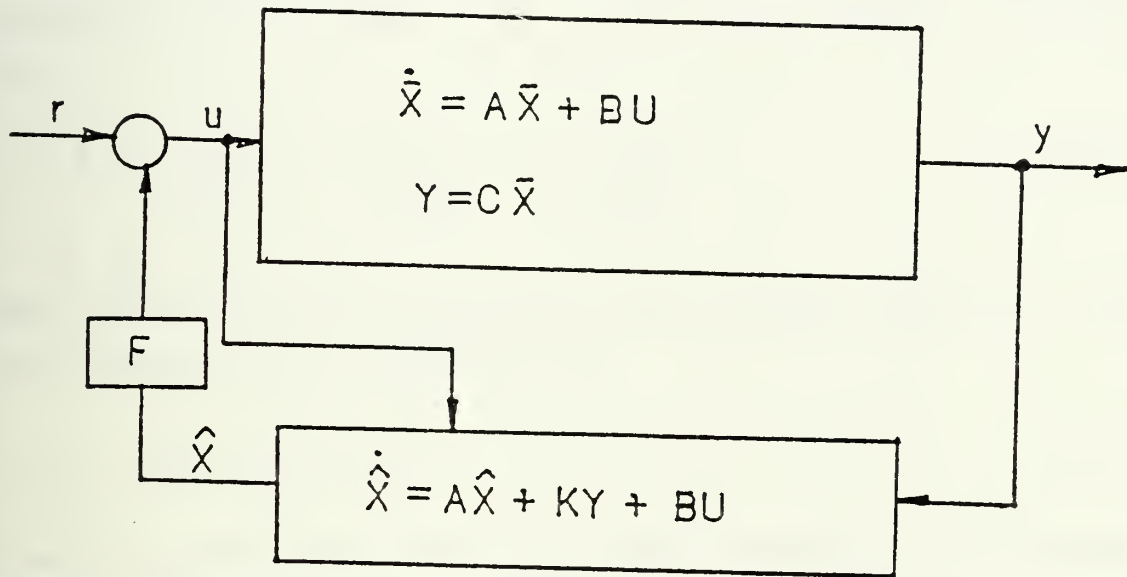


Figure 8.1 Simple Observer.

Given an observer as represented in figure 8.1 it has been shown that the system differential equation may be written as equation 8.1

$$\begin{bmatrix} \dot{\bar{x}} \\ \dot{\hat{x}} \end{bmatrix} = \begin{bmatrix} A & -BF \\ KC & A_c - BF \end{bmatrix} \begin{bmatrix} \bar{x} \\ \hat{x} \end{bmatrix} + \begin{bmatrix} B \\ B \end{bmatrix} r \quad (8.1)$$

$$A_c = A - KC$$

where \bar{x} is the state and \hat{x} is the observer variable, and the solution may be separated to independent solutions for the feedback gains and the observer gains. This separation allows the use of the feedback gains to set the pole

locations and then a second optimization run with set feedback gains so that the observer gains may be computed to adjust the system robustness level. The observer pole location could also be placed using the observer gains, K . In the currently implementation of the pole placement and robustness design routine the observer poles are simply restricted to areas of the left half plane.

In this chapter a simple stable observer system will be analyzed based on [Ref. 17]. Given the system, equation 8.2,

$$\begin{aligned} \dot{\underline{x}} &= \begin{bmatrix} 0 & 1 \\ -3 & -4 \end{bmatrix} \underline{x} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u + \begin{bmatrix} 35 \\ -61 \end{bmatrix} \underline{r} \\ y &= \begin{bmatrix} 2 & 1 \end{bmatrix} \underline{x} + \eta \end{aligned} \quad (8.2)$$

where $E(\underline{r}) = E(\eta) = 0$ and $E(\underline{r}(t)\underline{r}(\tau)) = E(\eta(t)\eta(\tau)) = \delta(t-\tau)$ with the feedback law of equation 8.3

$$u = -(50 \quad 10)x + 50r \quad (8.3)$$

An analysis has been done to compare results of the numerical optimization procedure with results presented in [Ref. 17]. The results for an optimal regulator design using quadratic cost criteria as stated in equation 8.4 are given in table 5

$$J = \int (\underline{x}^T H^T H \underline{x} + u^2) dt \quad (8.4)$$

with $H = 4\sqrt{5} \begin{pmatrix} \sqrt{35} & 1 \end{pmatrix}$

Figure 8.2 shows a Nyquist plot of the full state regulator, the optimal filter and a fast filter. The full state design had poles at $s = -7.0 \pm j 2.0$ and feedback gains of 50 and 10. The optimal filter as shown in table 5 had poles of $-7.0 \pm j 2.0$ with gain and phase margins of -6.75 db and ± 15 degrees. The optimum filter gains were 30 and -50. Using a faster

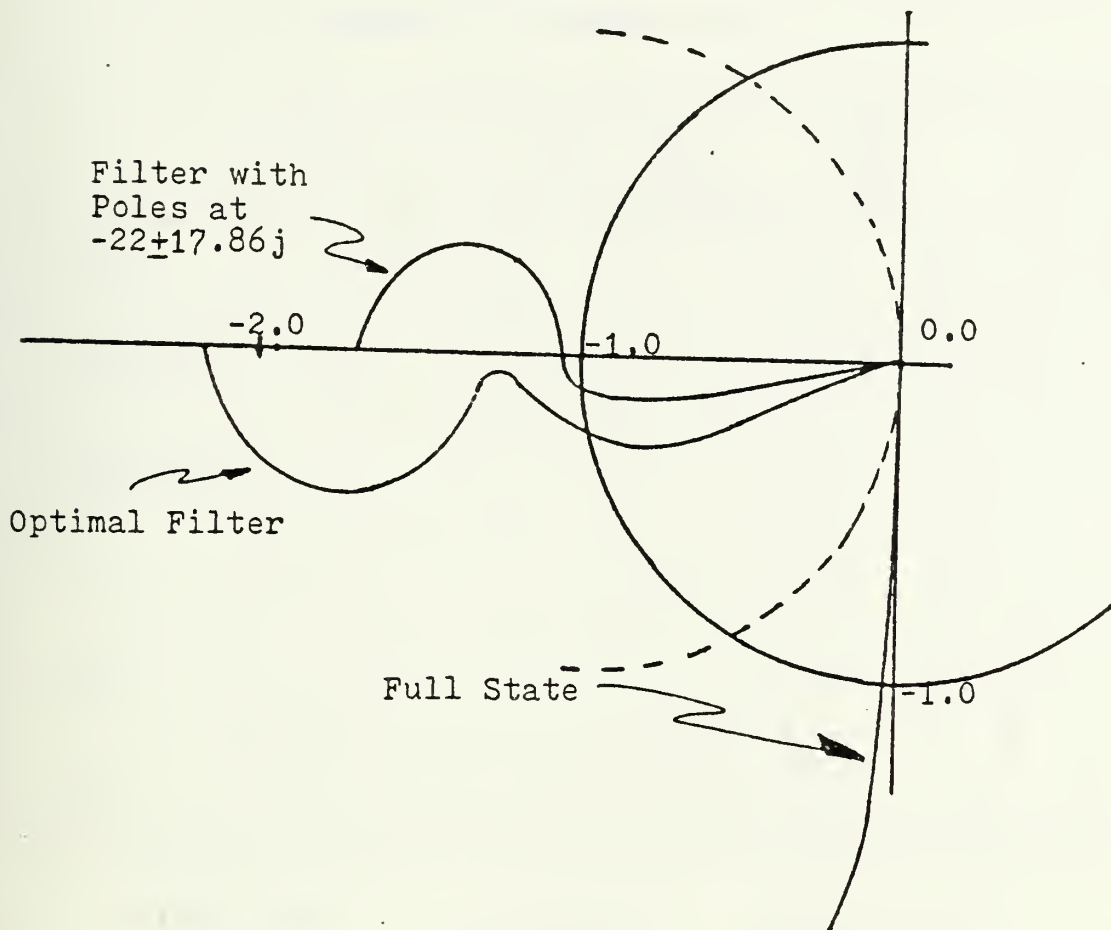


Figure 8.2 Nyquist Plot.

filter also gives poor gain and phase margins. The gain margins are on the order of -9.8 while the phase margin is less than 10 degrees. The bandwidth also increased from 12 to 40 rad/sec. A recovery procedure based on a modification to the process noise matrix [Ref. 17] may be applied to the problem. This procedure can recover a large amount of the robustness that was lost with the observer addition. Figure 8.3 shows data obtained for several trials of the fictitious noise procedure. The gain, phase margin and other parameters may be found in table 5.

TABLE 5
Observer Parameter Data

	FILTER POLES	GAIN MARGIN db	PHASE MARGIN deg	ERROR COVARIANCE $E[(x-\hat{x})(x-\hat{x})^T]$		STATE COVARIANCE $E(xx^T)$		FILTER GAIN
Optimal LQG Design	$-7 \pm 2j$	- 6.75	15	97 -163	-163 277	221 -613	-613 2070	30 -50
Fast Filter Adjust- ment Procedure	$-22 \pm 17.9j$	- .98	10	6280 -12200	-12200 23800	130 -613	-613 8520	720 -1400
Fictitious Noise Adjustment Procedure $q^2 = 100$	-8.3	- 7.73	19	107	-184	236	-613	26.8
	-13.1			-184	319	-613	1810	-40.2
	-2.9	-10.9	33	163	-301	268	-613	20.4
	-24			-301	564	-613	1500	-17.7
	-2.5	-13.9	42	204	-385	285	-613	16.7
	-33			-385	743	-613	1360	-1.8
	-2.1	-37	74	290	-570	317	-613	6.9
	-100			-570	1170	-613	1200	84.6

POPLAR CASE I	1.00018 -2.02742	GM=-5.5 dB PM=117 deg
POPLAR CASE II	18.22148 32.59476	GM=-19 dB PM=58 deg

The singular values of the state feedback and optimal observer systems were computed for comparison with results produced using the pole placement and robustness design recovery procedure. Figure 8.4 indicates a loss in robustness represented by the observer singular values. Lower singular values are less robust. For this single-input single-output observer the Nyquist diagram will be used to define gain and phase margins. Since a singular value of 1.0 is indicative of a linear quadratic level of gain and phase margin, i.e. GM=-6 db, and PM=±60 degrees, this was chosen

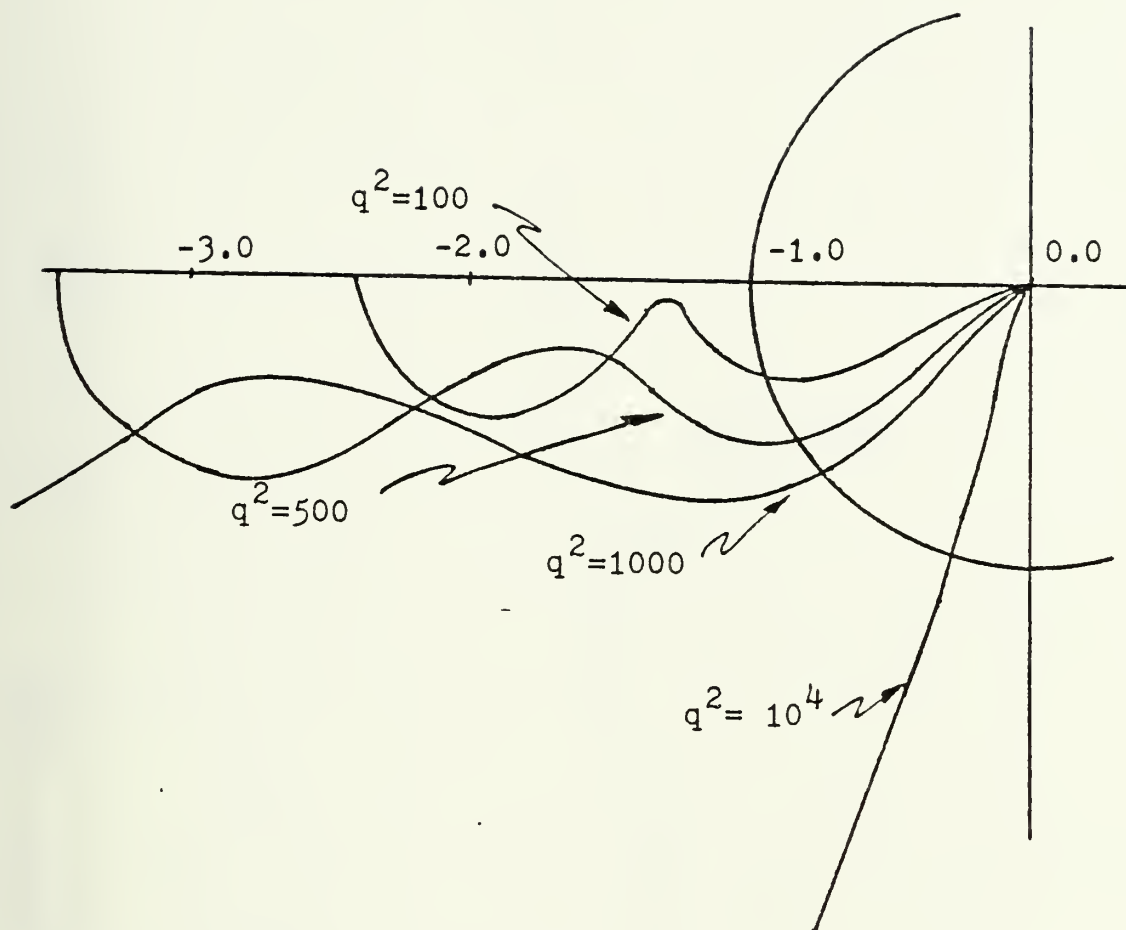


Figure 8.3 Nyquist Plot for Robustness Recovery.

as the design level. The pole placement and robustness routine was used to recover robustness while setting the pole placement at the state feedback pole locations of $-7 \pm j2$. The minimum input singular value level was set at 1.0. The pole placement and robustness routine was also set to place the observer poles anywhere between -100 and -2 that would provide robust design. The plant poles were placed at $-7.05 \pm j 1.82$. The feedback gains for this run

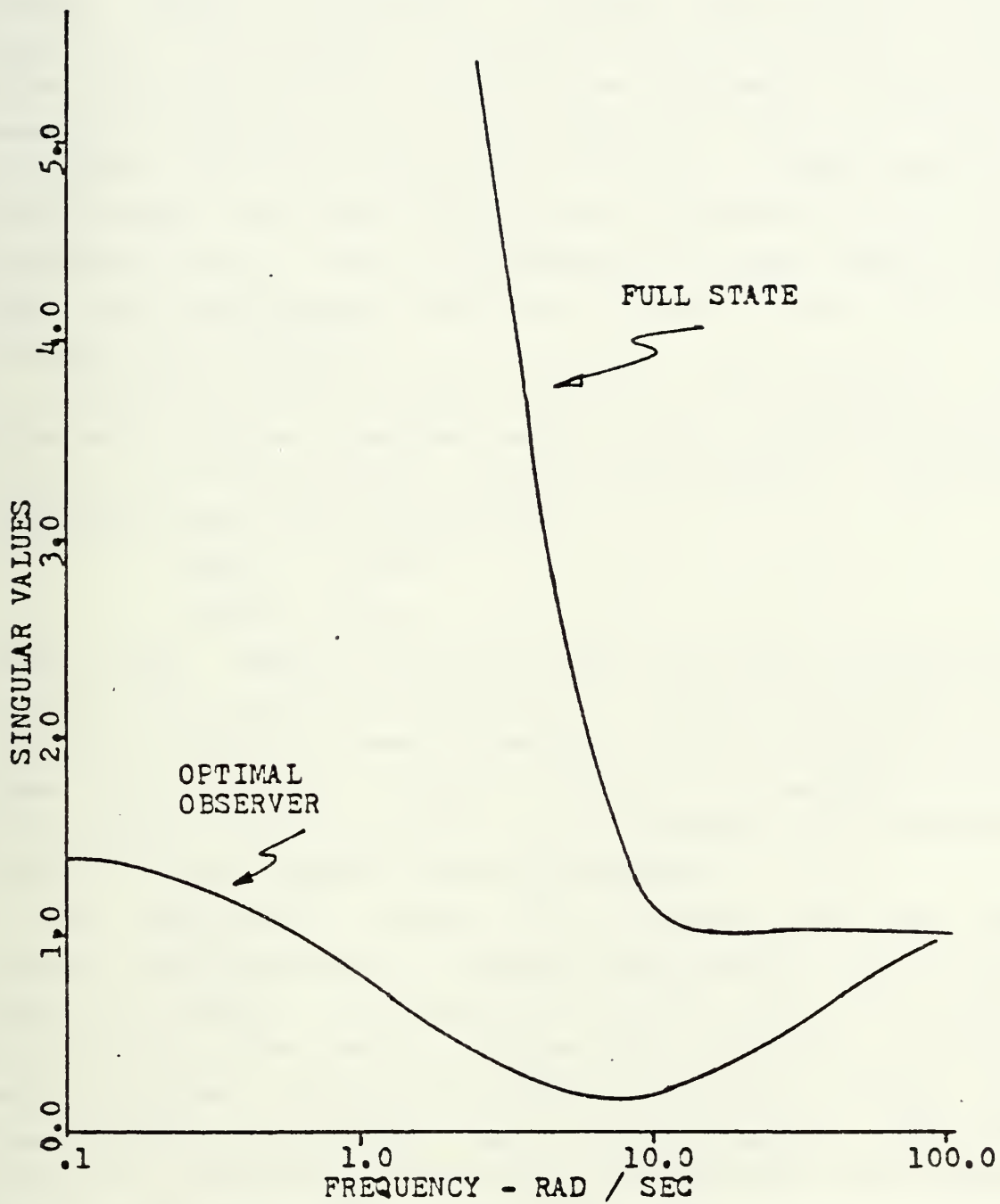


Figure 8.4 Singular Values of Observer System.

were 50 and 10.09. The optimizer output produced a filter with a gains of 1.00018 and -2.02742. The observer pole locations were $-1.99 \pm j .001$, very near the plant zero location. A significant change was produced in singular values. Figure 8.5 shows all singular value curves plotted together. The optimizer solution for this problem is well above the optimal filter curve at low frequency. Figure 8.6 shows the Nyquist plot of the optimizer developed design. The system has a gain margin of -6 db and a phase margin of 117 degrees. The most significant differences between the two designs being that the observer poles are close to the plant zero locations and the filter gains are much lower for the optimizer solution.

Using the OPTSYS program with the pole placement and robustness routine computed gains as the design parameters the data for the observer filter was computed. The error covariance matrix was found to be

$$\begin{bmatrix} 306.5 & -456.5 \\ -456.5 & 700.1 \end{bmatrix}$$

These values compare favorably with the trends established in table 5. The last comparison of the pole placement and robustness design recovery procedure was an analysis of the time response curves. Figure 8.7 shows the comparison plot. The design obtained using the pole placement and robustness routine did not degrade system performance.

One additional analysis was conducted that set the desired design parameters slightly differently. In this run the pole placement and robustness routine was set to place the observer poles between -10 and -100. The pole placement and robustness routine was unable to totally satisfy this requirement. It violated one of the constraints and moved the larger observer pole to -2.25 which is near the optimum pole location and also the high q^2 values of the fictitious noise procedure. The smaller pole was moved out to -70.8

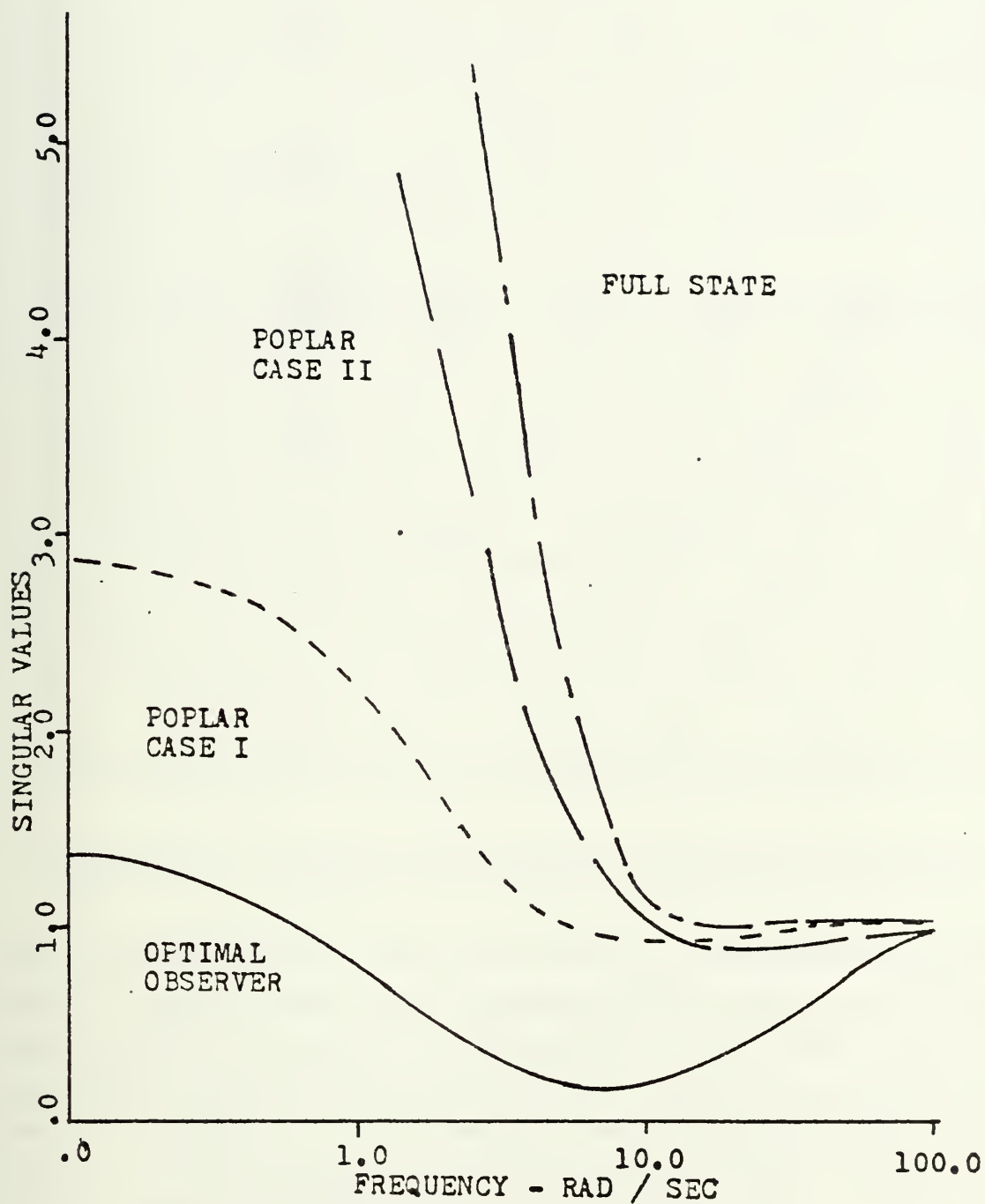


Figure 8.5 Singular Value Comparison Plot.

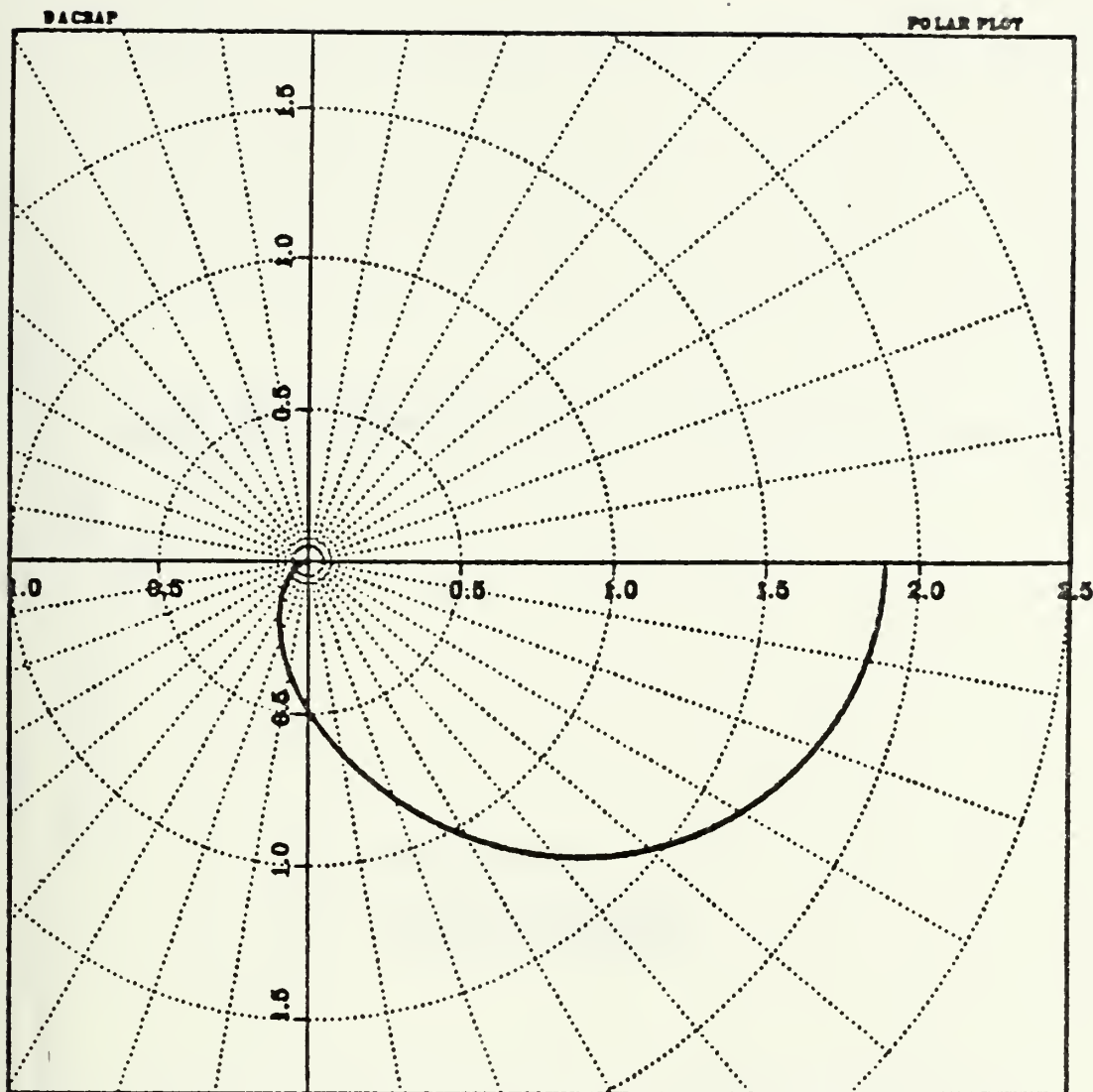


Figure 8.6 Nyquist for Computed Robustness Recovery.

which corresponds to the same level of pole movement found when q^2 becomes large. The singular value levels were raised significantly as shown in figure 8.5. The state and error covariance matrices were of the same order as the matrices found in table 5. This design has a phase margin of 58 degrees. The gain margin was about -19 db. Even though some of the constraint conditions on the design were not met the design demonstrates excellent robustness recovery.

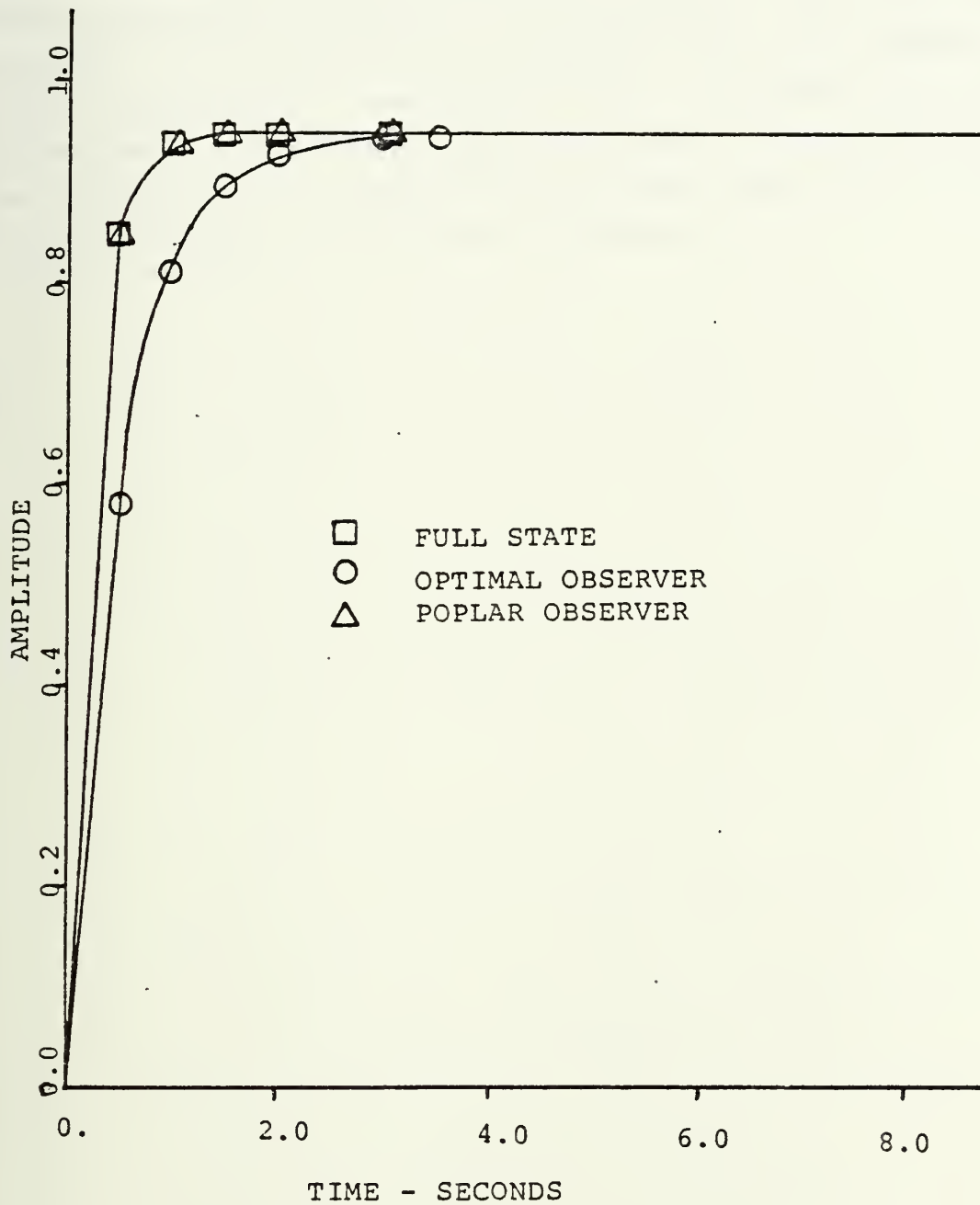


Figure 8.7 Time Response Plot for Simple Observer.

The pole placement and robustness design routine has produced a robust design for this observer based system. This design is obtained by using a numerical optimization to directly manipulate the feedback and filter gains. Modification of the IQ functional equation as done in the fictitious noise procedure is not required. The pole placement and robustness routine solution for this problem has lower gains than those found using the fictitious noise adjustment. The routine provides a good, direct methodology for selecting the feedback and filter gains for a robust observer design with excellent performance.

IX. ROBUST OBSERVER DESIGN

This chapter will be devoted to a short discussion of the robustness recovery of a fourth order observer based controller. The problem is a helicopter problem, [Ref. 6]. In this case the helicopter model is that of the longitudinal control loop of a CH-47. The nominal model is taken to be the system of equations 9.1 and 9.2 for an aircraft speed of forty knots.

$$\dot{\underline{x}} = \begin{bmatrix} -0.02 & 0.005 & 2.4 & -32. \\ -0.14 & 0.44 & -1.3 & -30. \\ 0. & 0.018 & -1.6 & 1.2 \\ 0. & 0. & 1. & 0. \end{bmatrix} \underline{x} + \begin{bmatrix} 0.14 & -.12 \\ .36 & -8.6 \\ .35 & .009 \\ 0. & 0. \end{bmatrix} \underline{u} \quad (9.1)$$

$$\underline{y} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 57.3 \end{bmatrix} \underline{x} \quad (9.2)$$

In this problem the controller is formulated as in figure 9.1 which leads to an open-loop transfer function of the form of equation 9.3

$$K(s)G(s) = F(sI - A + BF + KC)^{-1} KC(sI - A)^{-1} B \quad (9.3)$$

The pole placement and robustness recovery procedure was applied to this problem.

First, the standard full state feedback design was carried out using the Naval Postgraduate version of OPTISYS. This design produced excellent singular value output for the return difference as shown in figure 9.2. The lowest singular value being essentially 1, corresponding to a LQ design with -6 db to ∞ gain margin and 60 degrees of phase margin. The time response of the system was good as shown in

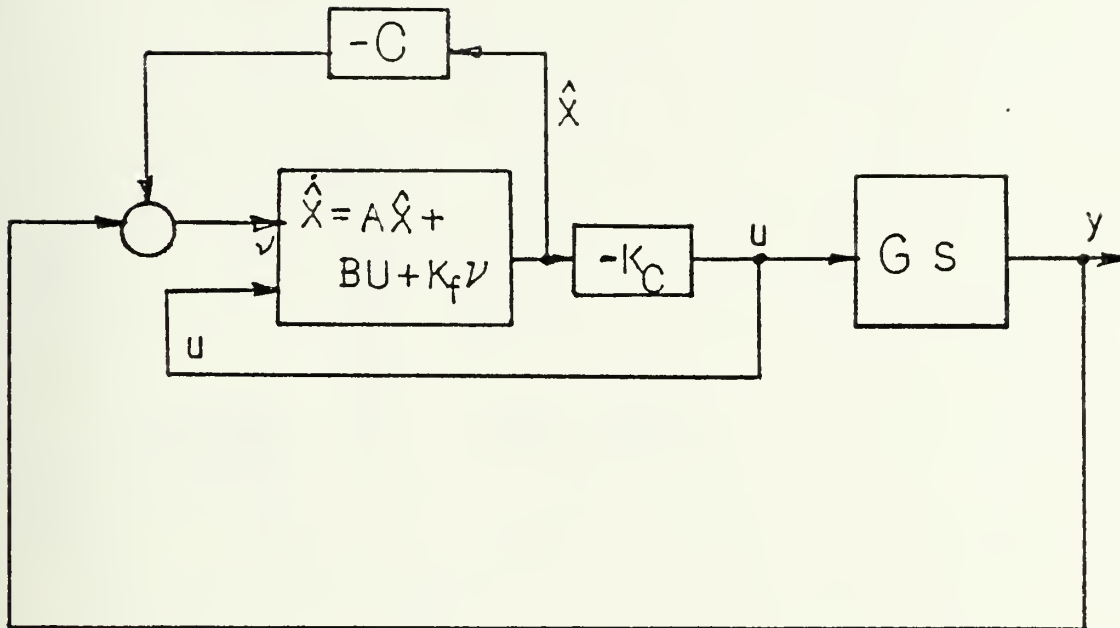


Figure 9.1 Observer Based Controller.

figure 9.3, reaching steady-state in about four seconds with only a slight overshoot. Since the characteristics for this system are acceptable no further design iteration was carried out. The full state feedback became the baseline design. Assuming that full state feedback was not available and only two measurements could be produced an observer was developed to control the two measured outputs, vertical velocity and pitch attitude. Using the measurement matrix of equation 9.2, OPTSYS was used to develop an optimal observer system for this problem. The singular values of the return difference matrix and the time response were plotted for comparison to the full state design. These are shown in figures 9.2 and 9.3 respectively. Note that the singular value of the return difference matrix is as low as 0.16 at 4 rad/sec. This equates to a gain margin of -1 db to 1 db and less than 10 degrees phase margin. The time response is

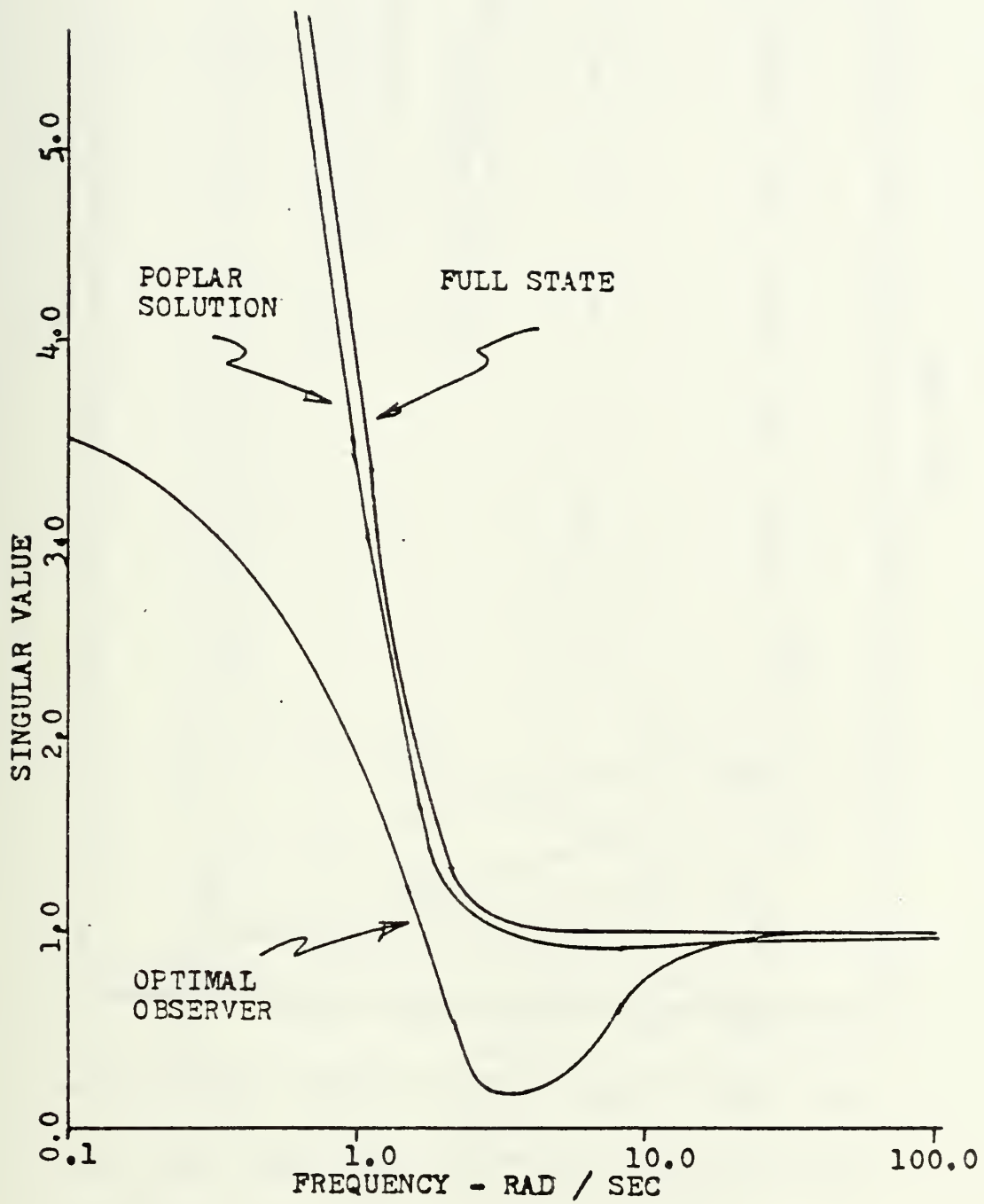


Figure 9.2 Singular Value Plot of Observer Results.

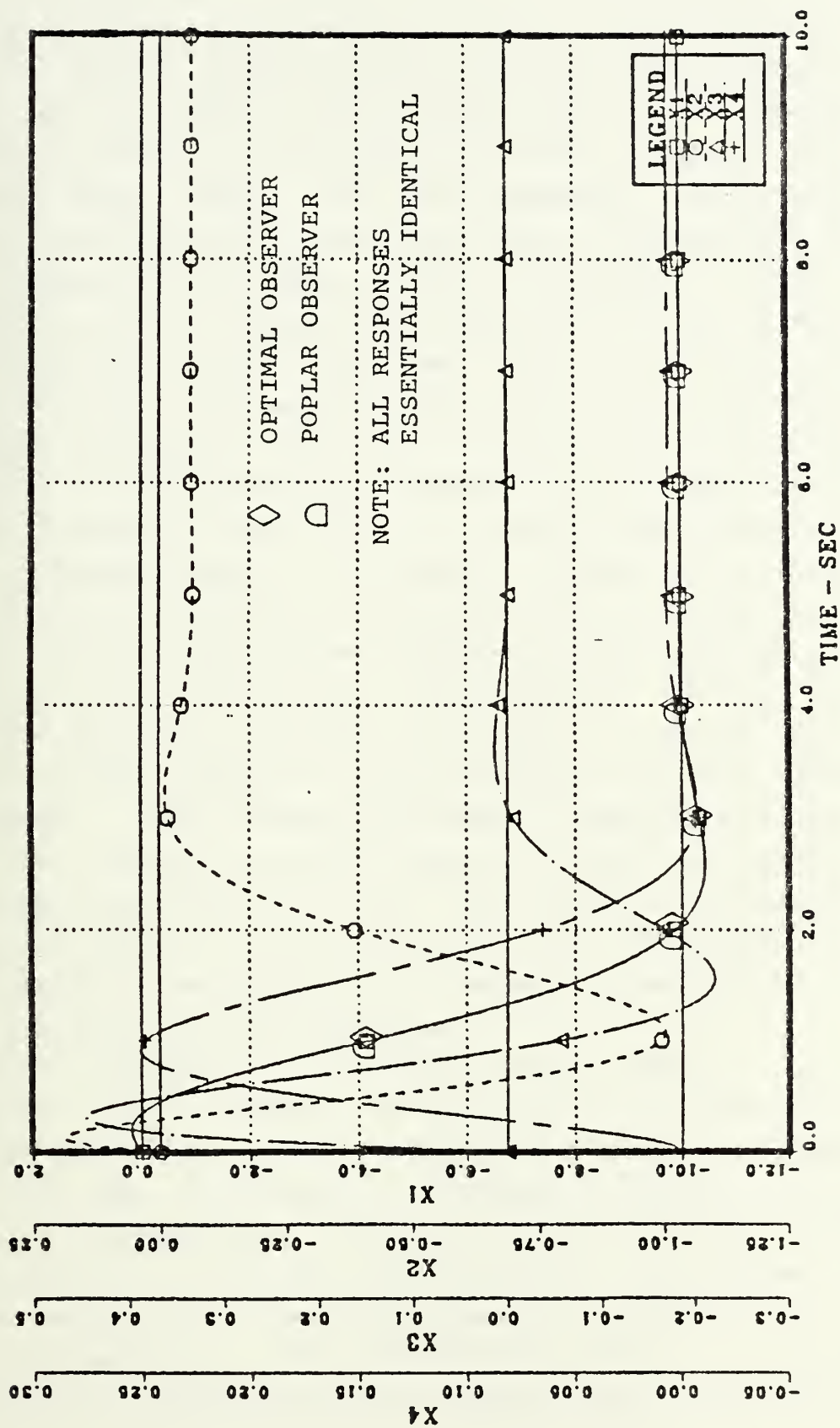


Figure 9.3 Time Response for Observer System.

plotted in figure 9.3. The steady-state was reached for the velocity after a slight overshoot in about four seconds.

As the final step in the analysis the pole placement and robustness routine was employed to recover robustness of the observer based system. The pole placement and robustness routine was first used to set the poles at approximately the same location as the poles of the LQ regulator. These were assumed to be the desired pole locations. The pole placement and robustness routine was then used to vary the filter gains, K , until the desired level of robustness was reached. The desired singular value level chosen was 1.0 which corresponds to -6 db to ∞ db gain margin and 60 degrees of phase margin. Figure 9.2 shows that the pole placement and robustness procedure failed to totally recover the robustness level to 1.0. The minimum singular value reached was only 0.936. This singular value equates to a gain margin of -5.0 db to ∞ and a phase margin of 55 degrees. While this is slightly less than the design objective, it is far superior to the optimal observer design discussed in the previous paragraphs. This recovery was made by making an optimization run, finding the filter gains with near zero values and freezing these values at zero. This reduced the number of design variables the pole placement and robustness routine was required to manipulate in a second optimization run and gave a higher robustness solution.

This analysis and the second order observer analysis presented earlier clearly indicate that it is possible to use the pole placement and robustness procedure and separation principle to develop a robustness recovery procedure. The pole placement and robustness routine provides robustness recovery by direct modification of the feedback and filter gains. This procedure requires no modification to LQ cost functionals or other parameters as done in the fictitious noise adjustment method commonly used for robustness

recovery. By providing direct gain adjustment the pole placement and robustness procedure results in a practical design with relatively low observer gains and good performance. The procedure is simple and straight forward with the only difficulty being a requirement to sometimes modify initial starting values or optimizer codes to force the solution toward the desired point.

X. CONCLUSIONS

An effective method of robustness multivariable control design utilizing a numerical optimization based algorithm has been developed. The pole placement and robustness design routine coupled with the Automated Design Synthesis program provides the designer an excellent tool with which to attack the robust design problem.

The pole placement and robustness design routine has demonstrated the capability of providing designs that solve the problems caused by cross-coupling perturbations which reduce robustness in multivariable systems. This design improvement is accomplished by modifying the system feedback gains in such a manner that the gain in channels that are affected by cross-coupling perturbations is equalized with other system gains to reduce this cross-coupling effect. The gain changes are accompanied by zero shifts which also influence the gain distribution and frequency response of the system.

Perturbation problems in multivariable systems have been shown to be detectable by singular value analysis and by using the Bode magnitude diagram of the open-loop transfer functions of the system. In the open loop transfer function large differentials in Bode gains and bandwidths are indicative of problem areas for cross-coupling perturbations. Robustness is obtained by the pole placement and robustness design program by modifying those gains and bandwidths associated with the cross-coupling perturbations thus reducing the amount of energy coupled from the perturbation into other channels. An associated zero shift has been observed when these gain modifications take place. This zero shift is in the direction of poles that are located in the vicinity

of the frequency of the minimum singular value and tends to equalize the frequency response curve gains in this region.

The use of numerical optimization to recover robustness in observer based designs was demonstrated. The pole placement and robustness routine was applied to problems previously solved using the fictitious noise procedure for robustness recovery. The direct manipulation of feedback and filter gains by the pole placement and robustness routine provided a highly robust design with relatively low filter gains. The problem of robustness recovery in filter-observer designs has been solved in a straight forward and highly practical manner.


```

*****
PPPPPP P OGCOC PPPPP L AAAAAA RRRRRR
P O O P P P P P A A R R
PPPPPP U C PPPPP L AAAAAA RRRRRR
P O C P P P P A A R R
P OCCOC P LLLLLL A A R R
*****
FOLE PLACEMENT AND ROBUSTNESS DESIGN PROGRAM
*****
THIS PROGRAM IS DESIGNED TO EMPLOY SINGULAR VALUE ANALYSIS AND
THE USE OF AN OPTIMIZATION ROUTINE TO AID IN POLE PLACEMENT
CONTROL SYSTEM DESIGN FOR LINEAR MULTIVARIABLE SYSTEMS.
ROBUSTNESS OF THE SYSTEM IS ALSO CONSIDERED BY ESTABLISHING
SINGULAR VALUE LEVELS WHICH CORRESPOND TO MULTILoop GAIN
AND PHASE MARGINS DETERMINED FROM THE UNIVERSAL GAIN PHASE
DIAGRAM DEVELOPED BY NEWSOM AND MUKHAPADHYAY AT NASA LANGLEY.
VERNON C GORDON, CDR, USN
VERSION 1.1, 8-10-84
*****
*****
THIS FIRST BLOCK SETS DIMENSIONS AND DECLARES REAL AND
COMPLEX FUNCTIONS. THE FOLLOWING IS A LIST OF PROGRAM SYMBOLS:
*****
*****
REAL VARIABLES BY NAME AND BRIEF DESCRIPTION
THE SYSTEM UNDER CONSIDERATION IS GIVEN AS:
XDOT = A*X + B*U; Y = C*X, FCR STATE OR U = -FY + R
A - THE PLANT A MATRIX (NSTATE X NSTATE)
B - THE PLANT B MATRIX (NSTATE X NSTATE)
C - THE PLANT C MATRIX (NSTATE X NSTATE)
F - THE PLANT OBSERV. OR OUTPUT MATRIX (NOUTPUT X NSTATE;
FOR STATE FEEDBACK PROBLEMS)
REALMU - THE FEEDBACK GAIN MATRIX (NCUMS)
IMAGMU - THE STATE FEEDBACK PROBLEMS)
FC - THE INPUT DESIRED REAL POLE LOCATION
BFC - THE INPUT DESIRED IMAGINARY POLE LOCATION
AMBFC - A - B*F*C
REIC - REAL COMPUTED EIGENVALUE OF THE SYSTEM
WK - PROGRAM WORK SPACE
WA - WORK SPACES USED BY SUBROUTINES
OMEGA - WORK DIMENSIONED WORK SPACE
SV - FREQUENCY SINGULAR VALUE OF (I + F * PLANT TRANSFER FUNCTION(G))
*****

```


[illegible]


```

** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** **  CCN01550
** ** ** **  CCN01540
** ** **  CCN01550
** ** **  CCN01560
** ** **  CCN01570
** ** **  CCN01580
** ** **  CCN01590
** ** **  CCN02000
** ** **  CCN02010
** ** **  CCN02020
** ** **  CCN02030
** ** **  CCN02040
** ** **  CCN02050
** ** **  CCN02060
** ** **  CCN02070
** ** **  CCN02080
** ** **  CCN02090
** ** **  CCN02100
** ** **  CCN02110
** ** **  CCN02120
** ** **  CCN02130
** ** **  CCN02140
** ** **  CCN02150
** ** **  CCN02160
** ** **  CCN02170
** ** **  CCN02180
** ** **  CCN02190
** ** **  CCN02200
** ** **  CCN02210
** ** **  CCN02220
** ** **  CCN02230
** ** **  CCN02240
** ** **  CCN02250
** ** **  CCN02260
** ** **  CCN02270
** ** **  CCN02280
** ** **  CCN02290
** ** **  CCN02300
** ** **  CCN02310
** ** **  CCN02320
** ** **  CCN02330
** ** **  CCN02340
** ** **  CCN02350
** ** **  CCN02360
** ** **  CCN02370
** ** **  CCN02380
** ** **  CCN02390
** ** **  CCN02400

NUMEER CF RCWS AND CULUMNS CF INPUT MATRICES AND
NUMBER OF PCLES TO BE INPUT FOR PLACEMENT
CARD 7 A MATRIX ; READ BY ROWS AS:
      A(1,1)      A(1,2)
      A(2,1)      A(2,2)
THERE WILL BE ONE CR MCRE CARDS FOR EACH ROW OF A
AFTER THE PROPER CARDS ARE READ FOR A THE NEXT MATRIX MUST
BE READ IN. THE REMAINDER OF THIS COMMENT WILL REFER TO EACH NEW
DATA SET AS A CARD. REMEMBER, THERE MAY BE SEVERAL CARDS PER
SET SO THE NUMBERS DON'T CORRESPOND DIRECTLY WITH THE DATA
LISTED IN THE INPUT FILE.
CARD 8 E MATRIX; READ ROW BY ROW
CARD 9 C MATRIX; READ ROW BY ROW
CARD 10 F MATRIX; READ ROW BY ROW
CARD 11 IF MATRIX; ROW BY ROW
      FGRMAT(81,2) ; THIS MATRIX DEFINES THE DESIGN VARIABLE
      TO THE OPTIMIZER. THE ENTRY WOULD BE OF THE FORM:
          1      2      3
THIS WOULD TELL THE OPTIMIZER TO MAKE F(1,1) THE FIRST
DESIGN VARIABLE (X(1)) AND SO FORTH. THE ZERO TELLS THE
OPTIMIZER THAT F(2,1) (IN THIS CASE) IS NOT A DESIGN
VARIABLE AND IS NOT TO BE CHANGED.
CARD 12 VLE,VUB; FORMAT(2F10.0)
      THERE WILL BE ONE CARD FOR EACH DESIGN VARIABLE
CARD 13 REALMU,IMAGMU ; FORMAT (2F10.0)
      INPUT REAL AND IMAGINARY DESIRED POLE LOCATION
      ONE POLE PER CARD
** ** ** **  *****
** ** **  INPUT RLA CODE AND PROGRAM CONTROL
** ** **  READ (1,750) KODE,KONIRL
** ** **  *****
** ** **  INPUT MAXN, INITIAL FREQUENCY,DELTA FREQ, OPTIMIZER WEIGHTS
** ** **  READ (1,710) WMAX,WI,DELTW
** ** **  READ (1,570) W1,W12,W13
** ** **  *****
** ** **  INPUT DESIRED GAIN AND PHASE MARGIN PARAMETERS
** ** **  READ (1,580) SVMINI,SVMINO,RJ,NIDG
** ** **  W=W1
** ** **  NC=1
** ** **  *****
** ** **  INPUT CFTIMIZER INPUT DATA (SEE ADS MANUAL)
** ** **  *****

```



```

C      READ (1,760) IGRAD,NDV,NCON,ISTRAT,IOPT,IONED,IPRINT,INFO
C
C      INPUT THE PARAMETER VALUES AND THE MATRICES A,B,C,F
C
C      READ (1,770) NROWA,NCCLA,NROWB,NCOLE,NROWC,NCOLC,NROWF,NCOLF,NMU
C      CALL READ (A,NROWA,NCOLA)
C      CALL READ (B,NROWB,NCOLB)
C      CALL READ (C,NROWC,NCOLC)
C      CALL READ (F,NROWF,NCOLF)
C
C      SPECIFY THE DESIGN VARIABLES WITHIN THE F MATRIX
C
C      DO 10 J=1,NROWF
C      READ (1,780) (IF(J,K),K=1,NCOLF)
C
C      SET THE DESIGN VARIABLE BOUNDS
C
C      DO 20 J=1,NDV
C      READ (1,790) VLB(J),VUB(J)
C
C      FORMULATE THE DESIGN VARIABLE X FROM THE INPUT GUESS FOR F
C
C      DO 30 J=1,NROWF
C      DO 40 K=1,NCOLF
C      IF (IF(J,K).EQ.0) GC TC 30
C      KK=IF(J,K)
C      X(KK)=F(J,K)
C      CONTINUE
C      CONTINUE
C      CONTINUE
C
C      READ THE DESIRED EIGENVALUES AS REAL(X)+REAL(Y)*I
C
C      DO 60 I=1,NMU
C      READ (1,860) REALMU(I),IMAGMU(I)
C      DO 70 J=1,NMU
C      MU(I)=CMPLX(REALMU(I),IMAGMU(I))
C      CONTINUE
C
C      SCRT THE INPUT EIGENVALUESUSING IMSL ROUTINE VSRTR
C
C      DO 80 I=1,NMU
C      IR(I)=I
C      CALL VSRTR (REALMU,NMU,IR)
C      DO 90 J=1,NMU
C      K=IR(J)
C      OML(J)=MU(K)
C      JJ=NMU-1
C
C      DO 2410
C      DO 2420
C      DO 2430
C      DO 2440
C      DO 2450
C      DO 2460
C      DO 2470
C      DO 2480
C      DO 2490
C      DO 2500
C      DO 2510
C      DO 2520
C      DO 2530
C      DO 2540
C      DO 2550
C      DO 2560
C      DO 2570
C      DO 2580
C      DO 2590
C      DO 2600
C      DO 2610
C      DO 2620
C      DO 2630
C      DO 2640
C      DO 2650
C      DO 2660
C      DO 2670
C      DO 2680
C      DO 2690
C      DO 2700
C      DO 2710
C      DO 2720
C      DO 2730
C      DO 2740
C      DO 2750
C      DO 2760
C      DO 2770
C      DO 2780
C      DO 2790
C      DO 2800
C      DO 2810
C      DO 2820
C      DO 2830
C      DO 2840
C      DO 2850
C      DO 2860
C      DO 2870
C      DO 2880

```



```

210 K=KEY(J)=EIG(K)
220 CEIG(J)=EIG(K)
      DO 210 KK=1,NROWA
      OZ(KK,J)=Z(KK,K)
      CONTINUE
      JJ=NCCLA-1
      DO 250 J=1,JJ
      IF (REAL(CEIG(J)),NE,REAL(CEIG(J+1))) GC TO 240
      IF (AIMAG(CEIG(J)).LT.AIMAG(CEIG(J+1))) GO TO 240
      TEMP1=CEIG(J)
      CEIG(J)=CEIG(J+1)
      CEIG(J+1)=TEMP1
      DO 230 KK=1,NROWA
      TEMP2=CZ(KK,J)
      OZ(KK,J)=OZ(KK,J+1)
      OZ(KK,J+1)=TEMP2
      CONTINUE
      CONTINUE
      CONTINUE
      *****
      CALL THE PLANT TRANSFER MATRIX ROUTINE AND FORM THE SYSTEM
      RETURN *****
      *****
      DEL=DELH
      KI=WI
      NC=1
      h=hi
      SMINAI=C.
      SMINMI=C.
      SMINAG=C.
      SMINMO=C.
      SMAXAI=C.
      SMAXMI=C.
      SMAXAO=C.
      SMAXMC=C.
      CALL PLANT (PLAN,AX,BX,CX,h,NRCWA,NCOLA,NROWB,NCOLB,NROWC,NCOLC)
      CALL CMATML (FX,PLAN,NROWF,NCOLF,NCCLB,FXPLT)
      CALL CMATML (PLAN,FX,NROWC,NCOLB,NCCLF,PLTFX)
      DO 280 J=1,NROWF
      DO 270 K=1,NCOLB
      XI(J,K)=C.0
      CONTINUE
      DO 290 J=1,NROWF
      XI(J,J)=1.0
      DO 310 J=1,NROWC

```

```

CCN03850
CCN03860
CCN03870
CCN03880
CCN03890
CCN03900
CCN03910
CCN03920
CCN03930
CCN03940
CCN03950
CCN03960
CCN03970
CCN03980
CCN03990
CCN04000
CCN04010
CCN04020
CCN04030
CCN04040
CCN04050
CCN04060
CCN04070
CCN04080
CCN04090
CCN04100
CCN04110
CCN04120
CCN04130
CCN04140
CCN04150
CCN04160
CCN04170
CCN04180
CCN04190
CCN04200
CCN04210
CCN04220
CCN04230
CCN04240
CCN04250
CCN04260
CCN04270
CCN04280
CCN04290
CCN04300
CCN04310
CCN04320

```



```

DO 300 I=1,NCOLF
XXI(J,K)=0.0
XXI(J,J)=1.0
XIPPFX(J,K)=XXI(J,K)+PLTFX(J,K)
CONTINUE
DO 310 I=1,NROWF
DO 320 K=1,NCOLB
XIPFPX(J,K)=XI(J,K)+FXPLI(J,K)
CONTINUE
DO 330 J=1,NROWF
DO 340 K=1,NROWF
FXPLTI(J,K)=0.0
FXPLTI(J,J)=1.0
CONTINUE
DO 350 J=1,NROWC
DO 360 K=1,NROWC
PLTFXI(J,K)=0.0
PLTFXI(J,J)=1.0
CONTINUE
CALL LECTIC (FXPLT,NROWF,10,FXPLTI,NROWF,10,0,WA,IER)
CALL LECTIC (PLTFX,NROWC,10,PLTFXI,NROWC,10,0,WA,IER)
DO 370 J=1,NROWF
DO 380 K=1,NROWC
XIFPLI(J,K)=XI(J,K)+FXPLTI(J,K)
CONTINUE
DO 390 J=1,NROWC
DO 400 K=1,NROWC
XIFPLI(J,K)=XXI(J,K)+PLTFXI(J,K)
CONTINUE
DO 410 J=1,NROWC
XIFPLI(J,K)=XXI(J,K)+PLTFXI(J,K)
CONTINUE
DO 420 J=1,NROWC
XIFPLI(J,K)=XXI(J,K)+PLTFXI(J,K)
CONTINUE
DO 430 J=1,NROWC
XIFPLI(J,K)=XXI(J,K)+PLTFXI(J,K)
CONTINUE
DO 440 J=1,NROWC
XIFPLI(J,K)=XXI(J,K)+PLTFXI(J,K)
CONTINUE
DO 450 J=1,NROWC
XIFPLI(J,K)=XXI(J,K)+PLTFXI(J,K)
CONTINUE
DO 460 J=1,NROWC
XIFPLI(J,K)=XXI(J,K)+PLTFXI(J,K)
CONTINUE
DO 470 J=1,NROWC
XIFPLI(J,K)=XXI(J,K)+PLTFXI(J,K)
CONTINUE
DO 480 J=1,NROWC
XIFPLI(J,K)=XXI(J,K)+PLTFXI(J,K)
CONTINUE
DO 490 J=1,NROWC
XIFPLI(J,K)=XXI(J,K)+PLTFXI(J,K)
CONTINUE
DO 500 J=1,NROWC
XIFPLI(J,K)=XXI(J,K)+PLTFXI(J,K)
CONTINUE
DO 510 J=1,NROWC
XIFPLI(J,K)=XXI(J,K)+PLTFXI(J,K)
CONTINUE
DO 520 J=1,NROWC
XIFPLI(J,K)=XXI(J,K)+PLTFXI(J,K)
CONTINUE
DO 530 J=1,NROWC
XIFPLI(J,K)=XXI(J,K)+PLTFXI(J,K)
CONTINUE
DO 540 J=1,NROWC
XIFPLI(J,K)=XXI(J,K)+PLTFXI(J,K)
CONTINUE
DO 550 J=1,NROWC
XIFPLI(J,K)=XXI(J,K)+PLTFXI(J,K)
CONTINUE
DO 560 J=1,NROWC
XIFPLI(J,K)=XXI(J,K)+PLTFXI(J,K)
CONTINUE
DO 570 J=1,NROWC
XIFPLI(J,K)=XXI(J,K)+PLTFXI(J,K)
CONTINUE
DO 580 J=1,NROWC
XIFPLI(J,K)=XXI(J,K)+PLTFXI(J,K)
CONTINUE
DO 590 J=1,NROWC
XIFPLI(J,K)=XXI(J,K)+PLTFXI(J,K)
CONTINUE
DO 600 J=1,NROWC
XIFPLI(J,K)=XXI(J,K)+PLTFXI(J,K)
CONTINUE
DO 610 J=1,NROWC
XIFPLI(J,K)=XXI(J,K)+PLTFXI(J,K)
CONTINUE
DO 620 J=1,NROWC
XIFPLI(J,K)=XXI(J,K)+PLTFXI(J,K)
CONTINUE
DO 630 J=1,NROWC
XIFPLI(J,K)=XXI(J,K)+PLTFXI(J,K)
CONTINUE
DO 640 J=1,NROWC
XIFPLI(J,K)=XXI(J,K)+PLTFXI(J,K)
CONTINUE
DO 650 J=1,NROWC
XIFPLI(J,K)=XXI(J,K)+PLTFXI(J,K)
CONTINUE
DO 660 J=1,NROWC
XIFPLI(J,K)=XXI(J,K)+PLTFXI(J,K)
CONTINUE
DO 670 J=1,NROWC
XIFPLI(J,K)=XXI(J,K)+PLTFXI(J,K)
CONTINUE
DO 680 J=1,NROWC
XIFPLI(J,K)=XXI(J,K)+PLTFXI(J,K)
CONTINUE
DO 690 J=1,NROWC
XIFPLI(J,K)=XXI(J,K)+PLTFXI(J,K)
CONTINUE
DO 700 J=1,NROWC
XIFPLI(J,K)=XXI(J,K)+PLTFXI(J,K)
CONTINUE
DO 710 J=1,NROWC
XIFPLI(J,K)=XXI(J,K)+PLTFXI(J,K)
CONTINUE
DO 720 J=1,NROWC
XIFPLI(J,K)=XXI(J,K)+PLTFXI(J,K)
CONTINUE
DO 730 J=1,NROWC
XIFPLI(J,K)=XXI(J,K)+PLTFXI(J,K)
CONTINUE
DO 740 J=1,NROWC
XIFPLI(J,K)=XXI(J,K)+PLTFXI(J,K)
CONTINUE
DO 750 J=1,NROWC
XIFPLI(J,K)=XXI(J,K)+PLTFXI(J,K)
CONTINUE
DO 760 J=1,NROWC
XIFPLI(J,K)=XXI(J,K)+PLTFXI(J,K)
CONTINUE
DO 770 J=1,NROWC
XIFPLI(J,K)=XXI(J,K)+PLTFXI(J,K)
CONTINUE
DO 780 J=1,NROWC
XIFPLI(J,K)=XXI(J,K)+PLTFXI(J,K)
CONTINUE
DO 790 J=1,NROWC
XIFPLI(J,K)=XXI(J,K)+PLTFXI(J,K)
CONTINUE
DO 800 J=1,NROWC
XIFPLI(J,K)=XXI(J,K)+PLTFXI(J,K)
CONTINUE

```



```
CN04810
CN04820
CN04830
CN04840
CN04850
CN04860
CN04870
CN04880
CN04890
CN04900
CN04910
CN04920
CN04930
CN04940
CN04950
CN04960
CN04970
CN04980
CN04990
CN05000
CN05010
CN05020
CN05030
CN05040
CN05050
CN05060
CN05070
CN05080
CN05090
CN05100
CN05110
CN05120
CN05130
CN05140
CN05150
CN05160
CN05170
CN05180
CN05190
CN05200
CN05210
CN05220
CN05230
CN05240
CN05250
CN05260
CN05270
CN05280
```

```
SVMI(NC)=SVMI(1)
SVMMO(NC)=SVMO(NROWC)
SVFXO(NC)=SVMO(1)
SIGPRO(NC)=SQRT((SIGNM1(NC)*SIGNM2(NC))
SMINAI=(SMINAI+(AMAX1(0.0, SVMINI-SIGNM1(NC))))**2)
SMINAO=(SMINAO+(AMAX1(0.0, SVMINO-SVADM(C(NC))))**2)

C      COMPUTE LCCF FOR ALL DESIRED FREQUENCY BANDWIDTH
C
C      W=WD+DEL
C      WD=10.C*WII
C      IF (W.LT.WD) GO TO 420
C      DEL=10.C*DEL
C      WII=10.C*WII
C      NC=NC+1
C      IF (W.LE.WMAX) GO TC 260
C      MAXA=NC-1

C      FORM QUANTITIES TO USE IN THE COST CRITERIA OF OPTIMIZER
C      SVMIN=SIGNM1(I)
C      DO 430 J=1,PAXN
C      IF (SIGNM1(J).GE.SVMIN) GO TO 430
C      SVMIN=SIGNM1(J)
C      CONTINUE
C      COST=0.
C      DO 440 J=1,NMU
C      COST=COST+((REAL(OMU(J))-REAL(CEIG(J)))**2)+(AIMAG(OMU(J))-AIMAG(CEIG(J)))**2)
C      CONTINUE

C      THE OBJECTIVE FUNCTION IS INSERTED HERE
C      OBJJ=WII*(COST+WT2/SVMIN)
C      OBJJ=WII*(COST+WT2*SMINAI+WT3*SMINAO)
C      OBJJ=WT2*SMINAI
C      IF (NCCA.EQ.O) GC TC 460
C      DU 450 J=1,NMU
C      IDG(J)=NIDG
C      R(J)=RJ
C      IGE(J)=RJ*(SQRT((REAL(CMU(J))-REAL(CEIG(J)))**2)+(AIMAG(CMU(J))-AIMAG(CEIG(J)))**2)
C      G(J)=SGFT1((REAL(CMU(J))-REAL(CEIG(J)))**2)+(AIMAG(CMU(J))-AIMAG(CEIG(J)))**2)
C      IG(J)=I#2)-R(J)
C      IF (KCENTFL.GT.O) GC TC 480
C      GO TO 120
C      CONTINUE

C      450
C      460
C      470
```



```

C THE GRADIENT COMPUTATION ROUTINE GOES HERE IF NEEDED
C GO TO 125
480 CONTINUE
C
C OUTPUT THE COMPUTED FINAL EIGEN VALUES AND VECTORS FROM OPT F'S
C
      CALL MMUL (F,C,NROWF,NCCLF,NCOLC,FC)
      CALL MMUL (B,FC,NROWB,NCOLE,NCCLC,BFC)
      DO 500 I=1,NROWA
      DO 490 J=1,NCOLA
      AMBFC(I,J)=A(I,J)-BFC(I,J)
      CONTINUE
      IF (KCNTRL.EQ.0) GC TC 510
      WRITE (6,62C)
      GO TO 520
      WRITE (6,63C)
      CALL EIGRF (AMBFC,NROWA,10,2,EIG,Z,10,WK,IER)
      WRITE (6,670)
      CALL CVECR (OEIG,NCOLA)
      DO 530 J=1,NMU
      WRITE (6,64C) OEIG(J)
      CALL CHFITE (OZ,NRCWA,NCOLA)
      WRITE (6,680)
      CALL WRITE (F,NROWF,NCOLF)
      WRITE (6,650) OBJ
C
C OUTPUT THE OPTIMIZED SINGULAR VALUE DATA
C
      WRITE (6,73C)
      WRITE (6,660) MAXN
      DO 540 J=1,MAXN
      WRITE (6,67C) OMEGA(J),SIGNM1(J),SIGNM2(J),SIGNMX(J),SIGPRG(J)
      WRITE (6,74C) OMEGA(J),SIGNM1(J),SIGNM2(J),SIGNMX(J),SIGPRO(J)
      DO 550 J=1,MAXN
      WRITE (6,68C)
      WRITE (6,69C) SVADM(J),SVADXU(J),SVMIM(J),SVMIM(J),SVMMD(J),SVMXD
      1(J)
      WRITE (6,700) OMEGA(J),SVACMO(J),SVADXC(J),SVMIM(J),SVMIX(J),SVMMD
      1(J),SVMXU(J)
      IF (KCNTRL.EQ.0.OR.KONTRL.GT.1) GO TO 560
      KONTRL=C
      GO TO 12C
      STOP
560 *****
C ***** STATEMENTS
C *****
570 *****
      FORMAT (3F10.0)

```



```

580 FORMAT (3F10.0,I5)
590 FORMAT (/,10H WEIGHT 1=,F10.5,9HWEIGHT 2=,F10.5,9HWEIGHT 3=,F10.5,
1/)
600 FORMAT (/,9H SVMINI=,F10.5,8H SVMINO=,F10.5,5H R J=,F10.5,6H IDG
1=,I5,/)
610 FORMAT (2I5)
620 FORMAT (/,28H *** THE INITIAL DATA IS ***,/)
630 FORMAT (/,41H *** THE OPTIMIZER COMPUTED OUTPUT IS ***,/)
640 FORMAT (2E12.5)
650 FORMAT (/,34H *** THE OBJECTIVE FUNCTION IS =,F10.5,6H ***,/)
660 FORMAT (I5)
670 FORMAT (5E12.5)
680 FORMAT (/,2X,7H FREQ ,5X,6H SVADMO,4X,6H SVALXG,4X,5H SVMIM,4X,5H SVM
1IX,4X,5H SVMMD,4X,5H SVMXO,/)
690 FORMAT (6E12.5)
700 FORMAT (7F10.0)
710 FORMAT (3F10.0)
720 FORMAT (/,6H WMAX=,F10.5,1X,10H INIT FREQ=,F10.5,1X,10H FREQ STEP=,F
110.5,/)
730 FORMAT (/,10H FREQUENCY,5X,13H MIN ADD IN SV,4X,13H 2ND ADD IN SV,3
1X,13H MAX ADD IN SV,2X,18H MIN PRCD ACC IN SV)
740 FORMAT (5(F10.5,5X))
750 FORMAT (I5,I5)
760 FORMAT (E15)
770 FORMAT (S12)
780 FORMAT (E12)
790 FORMAT (2F10.0)
800 FORMAT (1H1,/,/,12H <<<< RUN #,I5,6H >>>>,/)
810 FORMAT (27H ***,)
820 FORMAT (/,35H *** THE A PLANT MATRIX ***,/)
830 FORMAT (/,33H *** THE B CONTROL INPUT MATRIX ***,/)
840 FORMAT (/,38H *** THE C OBSERVATION MATRIX ***,/)
850 FORMAT (/,44H *** THE ORDERED COMPLEX EIGENVALUES (INPUT),/)
860 FORMAT (2F10.0)
870 FORMAT (/,37H *** THE ORDERED COMPUTED EIGENVALUES,/)
880 FORMAT (/,29H *** THE ORDERED EIGENVECTORS,/)
890 FORMAT (/,36H *** F FEEDBACK MATRIX ***,/)
END

```



```

10 CONTINUE
20 CONTINUE
30 CONTINUE
   RETURN
C ***
C *** REAL MATRIX MULTIPLICATION ROUTINE
C ***
C *** SUBROUTINE MMUL (R,T,M,LL,N,U)
C *** REAL*4 F(10,10),T(10,10),U(10,10)
C *** INTEGER M,LL,N
C *** DO 30 I=1,M
C *** DO 20 J=1,N
C *** U(I,J)=0.0
C *** DO 10 INDEX=1,LL
C *** U(I,J)=U(I,J)+R(I,INDEX)*T(INDEX,J)
C *** CONTINUE
C *** CONTINUE
C *** RETURN
C ***
C *** CSVD ALGORITHM FOR COMPLEX MATRICES: ANY OF THE ROUTINES OF IMSL
C *** CR LINPACK COULD ALSO BE USED WITH PROPER CAUTIONS
C ***
C *** SUBROUTINE CSVD (A,MMAX,NMAX,M,N,P,NU,NV,S,L,V)
C *** COMPLEX A(MMAX,1),L(MMAX,1),V(NMAX,1)
C *** INTEGER M,N,P,NU,NV
C *** REAL S(1)
C *** COMPLEX C,R
C *** REAL B(100),C(100),T(100)
C *** DATA ETA,TOL/1.5E-8,1.E-31/
C *** NP=N+P
C *** N1=N+1
C ***
C *** HCUSEHCLDEF REDUCTION
C *** C(1)=0.EC
C *** K=1
C *** K1=K+1
C ***
C *** ELIMINATION OF A(I,K), I=K+1, .....M
C *** Z=C.EO
C *** DO 20 I=K,M
C *** Z=Z+REAL(A(I,K))**2+AIMAG(A(I,K))**2
C *** B(K)=0.EC
C *** IF (Z.LE.TCL) GC TC 70
C *** Z=SQRT(Z)
C *** B(K)=Z

```



```

C      TOLERANCE FOR NEGLIGIBLE ELEMENTS
140  EPS=0.E0
    DO 150 K=1,N
      S(K)=B(K)
      T(K)=C(K)
150  EPS=AMAX1(EPS,S(K)+T(K))
      EPS=EPS*ETA
C
C      INITIALIZATION OF L AND V
    IF (NU.EC.O) GO TO 180
    DO 170 J=1,NU
      DO 160 I=1,M
160    U(I,J)=(0.E0,0.E0)
170    U(J,J)=(1.EC,0.E0)
180    IF (NV.EC.O) GO TO 210
      DO 200 J=1,NV
        DO 190 I=1,N
190      V(I,J)=(0.E0,0.E0)
200      V(J,J)=(1.EC,0.E0)
C
C      QR DIAGONALIZATION
210  DO 380 KK=1,N
      K=N1-KK
C
C      TEST FOR SPLIT
220  DO 230 LL=1,K
      L=K+1-LL
      IF (ABS(T(LL)).LE.EPS) GO TO 290
      IF (ABS(S(LL-1)).LE.EPS) GO TO 240
230  CONTINUE
C
C      CANCELLATION OF E(L)
240  CS=0.EC
      SN=1.E0
      LL=L-1
      J=L,K
      F=SN*T(I)
      T(I)=C*F+T(I)
      IF (ABS(F).LE.EPS) GO TO 290
      H=S(I)
      W=SQRT(F*F+H*H)
      S(I)=W
      CS=H/W
      SN=-F/W
      IF (NU.EC.O) GO TO 260
      DO 250 J=1,N
        X=REAL(U(J,LL))
        Y=REAL(U(J,I))

```

```

CWR01450
CWR01460
CWR01470
CWR01480
CWR01490
CWR01500
CWR01510
CWR01520
CWR01530
CWR01540
CWR01550
CWR01560
CWR01570
CWR01580
CWR01590
CWR01600
CWR01610
CWR01620
CWR01630
CWR01640
CWR01650
CWR01660
CWR01670
CWR01680
CWR01690
CWR01700
CWR01710
CWR01720
CWR01730
CWR01740
CWR01750
CWR01760
CWR01770
CWR01780
CWR01790
CWR01800
CWR01810
CWR01820
CWR01830
CWR01840
CWR01850
CWR01860
CWR01870
CWR01880
CWR01890
CWR01900
CWR01910
CWR01920

```



```

250 U(J,L1)=CMPLX(X*CS+Y*SN,0.E0)
260 U(J,I)=CMPLX(Y*CS-X*SN,0.E0)
    DO 270 J=N1,NP
    C=A(L1,J)
    R=A(I,J)
    A(L1,J)=C*CS+R*SN
    A(I,J)=R*CS-Q*SN
    CONTINUE
270
280 C
290 C
    TEST FOR CONVERGENCE
    W=S(K)
    IF (L.EC.K) GO TO 360
    C
    C
    CRIGIN SHIFT
    X=S(L)
    Y=S(K-1)
    G=T(K-1)
    H=T(K)
    F=((Y-W)*(Y+W)+(G-H)*(G+H))/(2.E0*H*Y)
    G=SQRT(F*F+1.E0)
    IF (F.L1.0.E0) G=-G
    F=((X-W)*(X+W)+(Y/(F+G)-H)*H)/X
    C
    C
    QR STEP
    CS=1.E0
    SN=1.EC
    L1=L+1
    DO 350 I=L1,K
    G=T(I)
    Y=S(I)
    H=SN*G
    G=CS*G
    W=SQRT(F*F+f*f)
    T(I-1)=W
    CS=F/W
    SN=H/W
    F=X*CS+Y*SN
    G=Y*CS-X*SN
    H=Y*SN
    Y=Y*CS
    IF (NV.EC.0) GO TO 310
    DO 300 J=1,N
    X=REAL(V(J,I-1))
    W=REAL(V(J,I))
    V(J,I-1)=CMPLX(X*CS+W*SN,0.E0)
    V(J,I)=CMPLX(W*CS-X*SN,0.E0)
    W=SQRT(H*H+f*f)
300
310

```



```

S(I-1)=W
CS=F/W
SN=H/W
F=CS*Y-SN*G
X=CS*Y-SN*G
IF (NU.EC.O) GO TO 330
DO 320 J=1,N
Y=REAL(U(J,I-1))
W=REAL(U(J,I))
U(J,I-1)=CMFLX(Y*CS+W*SN,O.EO)
U(J,I)=CMPLX(W*CS-Y*SN,O.EO)
IF (N.EC.NP) GO TO 350
DO 340 J=1,NP
Q=A(I-1,J)
R=A(I,J)
A(I-1,J)=Q*CS+R*SN
A(I,J)=R*CS-Q*SN
CONTINUE
T(L)=O.EO
T(K)=F
S(K)=X
GO TO 220

C CCNVERGENCE
360 IF (W.EE.O.EO) GC TC 380
S(K)=-W
IF (NV.EC.O) GO TO 380
DO 370 J=1,N
V(J,K)=-V(J,K)
CONTINUE
370
380 C SCRT SINGULAR VALUES
DO 450 K=1,N
G=-1.EC
J=K
DO 390 I=K,N
IF (S(I).LE.G) GO TC 390
G=S(I)
J=I
CONTINUE
390 IF (J.EC.K) GO TC 450
S(J)=S(K)
S(K)=G
IF (NV.EC.O) GO TO 410
DO 400 I=1,N
G=V(I,J)
V(I,J)=V(I,K)
V(I,K)=G
400

```

```

CWR02410
CWR02420
CWR02430
CWR02440
CWR02450
CWR02460
CWR02470
CWR02480
CWR02490
CWR02500
CWR02510
CWR02520
CWR02530
CWR02540
CWR02550
CWR02560
CWR02570
CWR02580
CWR02590
CWR02600
CWR02610
CWR02620
CWR02630
CWR02640
CWR02650
CWR02660
CWR02670
CWR02680
CWR02690
CWR02700
CWR02710
CWR02720
CWR02730
CWR02740
CWR02750
CWR02760
CWR02770
CWR02780
CWR02790
CWR02800
CWR02810
CWR02820
CWR02830
CWR02840
CWR02850
CWR02860
CWR02870
CWR02880

```



```

410 IF (NL-EC.O) GO TO 430
    DO 420 I=1,N
    U(I,J)=L(I,K)
420 U(I,K)=C
430 IF (N-EC.NP) GO TO 450
    DO 440 I=N1,NP
    C=A(J,I)
440 A(J,I)=A(K,I)
450 A(K,I)=C
    CONTINUE
C
C BACK TRANSFORMATION
    IF (NL-EC.O) GO TO 510
    DO 500 KK=1,N
    K=N1-KK
    IF (B(K).EQ.O.EO) GC TC 500
    Q=-A(K,K)/CABS(A(K,K))
    DO 460 I=1,NU
    U(K,J)=C*L(K,J)
460 DO 490 J=1,NU
    Q=(C-EC,C.EO)
    DO 470 I=K,M
    C=C+CC*JG(A(I,K))*U(I,J)
470 Q=C/(CABS(A(K,K))*B(K))
    DO 480 I=K,M
    L(I,J)=L(I,J)-Q*A(I,K)
480 CONTINUE
490 CONTINUE
500 IF (NV-EC.O) GO TO 570
510 IF (N-LT.2) GO TO 570
    DO 560 KK=2,N
    K=N1-KK
    K1=K+1
    IF (C(K1).EQ.O.EO) GO TO 560
    Q=-CONJG(A(K,K1))/CABS(A(K,K1))
    DO 520 I=1,NV
    V(K1,J)=C*V(K1,J)
520 DO 550 J=1,NV
    C=(O.EO,O.EO)
    DO 530 I=K1,N
    Q=C+A(K,I)*V(I,J)
530 Q=C/(CABS(A(K,K1))*C(K1))
    DO 540 I=K1,N
    V(I,J)=V(I,J)-Q*CONJG(A(K,I))
540 CONTINUE
550 CONTINUE
560 RETURN
570

```

```

CWR02590
CWR02500
CWR02510
CWR02520
CWR02530
CWR02540
CWR02550
CWR02560
CWR02570
CWR02580
CWR02590
CWR03000
CWR03010
CWR03020
CWR03030
CWR03040
CWR03050
CWR03060
CWR03070
CWR03080
CWR03090
CWR03100
CWR03110
CWR03120
CWR03130
CWR03140
CWR03150
CWR03160
CWR03170
CWR03180
CWR03190
CWR03200
CWR03210
CWR03220
CWR03230
CWR03240
CWR03250
CWR03260
CWR03270
CWR03280
CWR03290
CWR03300
CWR03310
CWR03320
CWR03330
CWR03340
CWR03350
CWR03360

```



```

C *** ENC *** C *** 370
C *** READ / REAL MATRIX *** C *** 380
C *** SUBROUTINE READ (A,N,K) *** C *** 390
C *** REAL*4 A(10,10) *** C *** 400
C *** SUBROUTINE READ (A,N,K) *** C *** 410
C *** REAL*4 A(10,10) *** C *** 420
C *** INTEGER N,K *** C *** 430
C *** DO 10 I=1,N *** C *** 440
C *** READ (1,20) (A(I,J),J=1,K) *** C *** 450
C *** CONTINUE *** C *** 460
C *** RETURN *** C *** 470
C *** FORMAT (EF10.0) *** C *** 480
C *** ENC *** C *** 490
C *** WRITE / REAL MATRIX *** C *** 500
C *** SUBROUTINE WRITE (A,N,K) *** C *** 510
C *** REAL*4 A(10,10) *** C *** 520
C *** SUBROUTINE WRITE (A,N,K) *** C *** 530
C *** REAL*4 A(10,10) *** C *** 540
C *** INTEGER N,K *** C *** 550
C *** DO 10 I=1,N *** C *** 560
C *** WRITE (6,20) (A(I,J),J=1,K) *** C *** 570
C *** CONTINUE *** C *** 580
C *** RETURN *** C *** 590
C *** FORMAT (EF10.5) *** C *** 600
C *** ENC *** C *** 610
C *** CONVERT REAL MATRIX TO COMPLEX FORM *** C *** 620
C *** SUBROUTINE CPLXCV (X,A,N,K) *** C *** 630
C *** COMPLEX*8 X(10,10) *** C *** 640
C *** SUBROUTINE CPLXCV (X,A,N,K) *** C *** 650
C *** COMPLEX*8 X(10,10) *** C *** 660
C *** REAL*4 A(10,10) *** C *** 670
C *** INTEGER N,K *** C *** 680
C *** DO 20 I=1,N *** C *** 690
C *** DO 10 J=1,K *** C *** 700
C *** X(I,J)=CMPLX(A(I,J),0.00) *** C *** 710
C *** CONTINUE *** C *** 720
C *** RETURN *** C *** 730
C *** ENC *** C *** 740
C *** THIS ROUTINE WILL COMPUTE THE TRANSFER FUNCTION AS A *** C *** 750
C *** FUNCTION OF FREQUENCY FOR A GIVEN A, B, C SET OF *** C *** 760
C *** MATRICES *** C *** 770
C *** SUBROUTINE FLANT (GP,AX,BX,CX,h,NRA,NCA,NRB,NCB,NRC,NCC) *** C *** 780
C *** COMPLEX*8 CX(10,10),AX(10,10),BX(10,10),GP(10,10),I(10,10),SIA(10, *** C *** 790
C *** 110),SIAI(10,10),SIAIB(10,10),SI(10,10) *** C *** 800
C *** COMPLEX*8 *** C *** 810
C *** 110),SIAI(10,10),SIAIB(10,10),SI(10,10) *** C *** 820
C *** COMPLEX*8 *** C *** 830
C *** 110),SIAI(10,10),SIAIB(10,10),SI(10,10) *** C *** 840

```



```

REAL*4 WA(100)
S=CMPLX(C.0,W)
DO 20 J=1,NRA
DO 10 K=1,NCA
SI(J,K)=C.0
I(J,K)=C.0
CONTINUE
DO 30 J=1,NCA
I(J,J)=1.0
SI(J,J)=C*I(J,J)
DO 50 J=1,NRA
DO 40 K=1,NCA
SIA(J,K)=SI(J,K)-AX(J,K)
CONTINUE
CALL LECTILC (SIA,NCA,10,I,NCA,10.0,WA,IER)
CALL CMAIML (I,BX,NCA,NCA,NCB,SIAIB)
CALL CMAIML (CX,SIAIB,NRC,NCC,NCB,GP)
RETURN
END
C *****
C ***** IS INPUT FOR DE SCLVER *****
C ***** DE SOLVER IS ADDED DIRECTLY TO CONXSV *****
C *****
C ***** SUBROUTINE FCN (N,T,X,XDOT) *****
REAL*4 T,X(N),XDOT(N),AMBFC(10,10),A(10,10),B(10,10),C(10,10),
1F(10,10),PHI(10),BFC(10,10),FC(10,10),AMBFCX(10),BPHI(10),
COMMON AMBFC,A,B,F,C,PHI,AMBFCX,BPHI,NCCLB
WRITE(6,4004)N,NCOLB
C4004 FORMAT(2I5)
C *****
C ***** WRITE(6,4006)(PHI(J),J=1,NCOLB) *****
C4006 FORMAT(2F10.5)
C *****
C ***** CALL WRITE(B,N,NCOLB) *****
C ***** CALL WRITE(AMBFC,N,N) *****
DO 10 J=1,N
BPHI(J)=0.0
DO 20 K=1,N
BPHI(J)=BPHI(J)+B(J,K)*PHI(K)
CONTINUE
C *****
C ***** WRITE(6,5500)(BPHI(J),J=1,N) *****
C5500 FORMAT(6F10.5)
DO 30 J=1,N
AMBFCX(J)=0.0
DO 40 K=1,N
AMEBFCX(J)=AMBFCX(J)+AMBFC(J,K)*X(K)
CONTINUE
C *****
C ***** WRITE(6,5501)(AMBFCX(J),J=1,N) *****
C5501 FORMAT(6F10.5)
DO 50 J=1,N

```



```

5C XDCT(J)=AMBFCX(J)+BPHI(J)
C WRITE(6,5502)(XDOT(J),J=1,N)
C5502 FORMAT(1F10.5)
C RETURN
END
C THIS SUBROUTINE COMPUTES THE CONTROLLER TRANSFER FUNCTION
C FOR THE CBSEFVER ROBUSTNESS PROBLEM
C
C
      SUBROUTINE CONTRL(H,AX,FX,FKX,FKC,BX,NRA,NCA,NRF,NCF,NRK,NCK,M,
1NRE)
      COMPLEX*8 H(10,10),AX(10,10),FX(10,10),FKX(10,10),FKC(10,10),
      *BX(10,10),SI(10,10),I(10,10),SIA(10,10),AALG(10,10),AALK(10,10)
1BFX(10,10),KA(100),FKC(10,10),BF(10,10)
      REAL*4 KA(100),FKC(10,10),BF(10,10)
      COMPLEX*8
      S=CMPLX(0.0,M)
      CALL CWRITE(AX,NRA,NCA)
      CALL CWRITE(FX,NRF,NCF)
      CALL CWRITE(FKX,NRK,NCK)
      DO 20 J=1,NRA
      DO 10 K=1,NCA
1C SI(J,K)=C.
2C CONTINUE
      DO 12 K=1,NRA
      DO 11 J=1,NCA
11 I(J,K)=C.0
12 CONTINUE
      DO 30 J=1,NCA
      DO 31 J=1,NRA
3C I(J,J)=1.0
      DO 32 K=1,NCA
      DO 31 J=1,NRA
      SI(J,J)=S*I(J,J)
32 CONTINUE
31 CALL CWRITE(SI,NRA,NCA)
      DO 50 J=1,NRA
      DO 40 K=1,NCA
      SIA(J,K)=SI(J,K)-AX(J,K)
40 CONTINUE
5C CONTINUE
      CALL CWRITE(SIA,NRA,NCA)
      CALL CFLXCV(FKX,FKC,NRA,NCA)
      CALL CNATML(BFX,BX,FX,NRB,NCF)
      DO 60 J=1,NRA
      DO 70 K=1,NCA
7C AALG(J,K)=SIA(J,K)+FKCX(J,K)+BFX(J,K)
      CALL CWRITE(AAUG,NRA,NCA)
8C CONTINUE
      CALL LECTIC(AAUG,NCA,10,I,NCA,10,0,KA,IER)

```

CWR044330
CWR044340
CWR044350
CWR044360
CWR044370
CWR044380
CWR044390
CWR044400
CWR044410
CWR044420
CWR044430
CWR044440
CWR044450
CWR044460
CWR044470
CWR044480
CWR044490
CWR044500
CWR044510
CWR044520
CWR044530
CWR044540
CWR044550
CWR044560
CWR044570
CWR044580
CWR044590
CWR044600
CWR044610
CWR044620
CWR044630
CWR044640
CWR044650
CWR044660
CWR044670
CWR044680
CWR044690
CWR044700
CWR044710
CWR044720
CWR044730
CWR044740
CWR044750
CWR044760
CWR044770
CWR044780
CWR044790
CWR044800

C
CALL CMAITML(I,FKX,NRA,ACA,NCK,AAUK)
CALL CMAITML(FX,AAUK,NRF,NCF,NCK,H)
CALL CMFITE(H,NRF,NCK)
RETURN
END

CAR04810
CWR04820
CWR04830
CWR04840
CWR04850


```

C      CALL READ (C,NROWC,NCCLC)
C      CALL READ (F,NROWF,NCOLF)
C      CALL READ (FK,NROWK,NCCLK)
C
C      SPECIFY THE DESIGN VARIABLES WITHIN THE F MATRIX
C
C      DO 10 J=1,NROWF
C      READ (1,1070) (IF(J,K),K=1,NCOLF)
C      DO 20 J=1,NROWK
C      READ (1,1010) (IFK(J,K),K=1,NCOLK)
C
C      SET THE DESIGN VARIABLE BOUNDS
C
C      DO 30 J=1,NLV
C      READ (1,1080) VLB(J),VLB(J)
C      DO 60 J=1,NROWF
C      DO 50 K=1,NCOLF
C      IF (IF(J,K).EQ.0) GO TO 40
C      KK=IF(J,K)
C      X(KK)=F(J,K)
C      CONTINUE
C      CONTINUE
C      CONTINUE
C
C      SPEC DESIGN VAR WITHIN THE FK MATRIX
C
C      READ THE DESIRED EIGENVALUES AS REAL(X)+REAL(Y)*I
C      DO 70 I=1,NMU
C      READ (1,1150) REALMU(I),IMAGMU(I)
C      DO 80 I=1,NMU
C      MU(I)=CMPLX(REALMU(I),IMAGMU(I))
C      CONTINUE
C
C      SORT THE INPUT EIGEN VALUES
C
C      DO 90 I=1,NMU
C      IR(I)=1
C      CALL VSFTR (REALMU,NMU,IR)
C      DO 100 J=1,NMU
C      K=IR(J)
C      OMU(J)=MU(K)
C      JJ=NMU-1
C      DO 120 J=1,JJ
C      IF (REAL(CMU(J)).NE.REAL(OMU(J+1))) GO TO 110
C      IF (AIMAG(CMU(J)).LT.AIMAG(OMU(J+1))) GO TO 110
C      TEMPL=CMU(J)
C      CMU(J)=CMU(J+1)

```



```

270      CO 270      J=1,NMU
      COST=CCST+((REAL(OMU(J))-REAL(CEIG(J)))*2+(AIMAG(OMU(J))-AIMAG(CE
      IG(J)))*2)
      CONTINUE
      CBJ=WT1*CCST
      IF (NCCA.EQ.0) GO TO 290
      DO 280      J=1,NMU
      IDG(J)=AIDG
      R(J)=RJ
      R(J)=RJ*SQRT((REAL(CMU(J))-REAL(CEIG(J)))*2+(AIMAG(OMU(J))-AIMAG(
      OEIG(J)))*2)
      G(J)=SQRT((REAL(CMU(J))-REAL(CEIG(J)))*2+(AIMAG(OMU(J))-AIMAG(CEI
      G(J)))*2)-R(J)
      IF (KCNTRL.GT.0) GO TO 310
      GO TO 130
      CONTINUE
      C THE GRADIENT COMPUTATION ROUTINE GOES HERE IF NEEDED
      C
310      CONTINUE
      WRITE (6,1160)
      CALL CVCWR (OEIG,NCOLA)
      DO 320      J=1,NMU
      WRITE (6,1170) OEIG(J),EIGK(J)
      CALL CHFINITE (OZ,NROWA,NCOLA)
      WRITE (6,1180)
      CALL WRITE (F,NRCWF,NCCLF)
      REAL (1,1050) IGRAC,NCV,NCCN,ISTRAT,IOPT,IONED,IPRINT,INFO
      DO 330      J=1,NDV
      READ (1,1080) VLBI(J),VUBI(J)
      DO 360      J=1,NROWK
      DO 350      K=1,NCOLK
      IF (IFK(J,K).EQ.0) GO TO 340
      KK K=IFK(J,K)+KK
      XI(KKK)=FK(J,K)
      CONTINUE
      CONTINUE
      IF (KCNTRL.GT.0) GO TO 141
      INFC=-2
      *****
      CALL ADS ROUTINE TO FIND FILTER FEEDBACK GAINS FCR
      THE ROBUSTNESS RECOVERY WITH THE CBSERVER GAINS
      *****
      CALL ACS (INFO,ISTRAT,IOPT,IONED,IFPRINT,IGRAD,ADV,NCUN,X1,VLBI,VUB
      I,CEBI,CI,ICG,NGT,IC,DF,XA,NRA,NCA,WK1,NRWK,IWK,NR IWK)
      IWK(2)=1
      *****

```



```

1COLK,K)
CALL CMATML (CONT,PLAN,NROWF,NCOLK,NCOLE,HG)
CALL CMATML (PLAN,CENT,NROWC,NCOLB,NCOLK,GH)
DO 450 J=1,NROWF
DO 440 K=1,NCOLB
XI(J,K)=0.0
CONTINUE
DO 460 J=1,NROWF
XI(J,J)=1.0
DO 480 J=1,NROWC
DO 470 K=1,NCOLK
XXI(J,K)=0.0
CONTINUE
DO 490 J=1,NCOLK
XXI(J,J)=1.0
DO 510 J=1,NROWF
DO 500 K=1,NCOLB
XIPHG(J,K)=HG(J,K)
XIPHG(J,K)=XXI(J,K)+HG(J,K)
CONTINUE
DO 530 J=1,NROWC
DO 520 K=1,NCOLK
XIPGH(J,K)=XXI(J,K)+GH(J,K)
CONTINUE
CONTINUE
DO 550 J=1,NROWF
HG I(J,K)=0.0
HG I(J,J)=1.0
CONTINUE
DO 570 J=1,NROWC
DO 560 K=1,NCOLK
GHI(J,K)=0.0
GHI(J,J)=1.
CONTINUE
CALL LEGTIC (HG,NROWF,10,HGI,NROWF,10,0,WA,IER)
CALL LEGTIC (GH,NROWC,10,GHI,NROWC,10,0,WA,IER)
DO 590 J=1,NROWF
DO 580 K=1,NCOLB
XIPHGI(J,K)=XI(J,K)+HGI(J,K)
CONTINUE
CONTINUE
DO 610 J=1,NROWC
DO 600 K=1,NCOLK
XIPGHI(J,K)=XXI(J,K)+GHI(J,K)
CONTINUE
CONTINUE
600
610
C500
500
510
520
530
540
550
560
570
C
580
590
600
610
CCN05290
CCN05300
CCN05310
CCN05320
CCN05330
CCN05340
CCN05350
CCN05360
CCN05370
CCN05380
CCN05390
CCN05400
CCN05410
CCN05420
CCN05430
CCN05440
CCN05450
CCN05460
CCN05470
CCN05480
CCN05490
CCN05500
CCN05510
CCN05520
CCN05530
CCN05540
CCN05550
CCN05560
CCN05570
CCN05580
CCN05590
CCN05600
CCN05610
CCN05620
CCN05630
CCN05640
CCN05650
CCN05660
CCN05670
CCN05680
CCN05690
CCN05700
CCN05710
CCN05720
CCN05730
CCN05740
CCN05750
CCN05760

```



```

C      GO SINGULAR VALUE DECCMP AND QUANTIFY ALL DESIRED SV'S
C
C      CALL CSVC (XIPHG,10,10,NROWF,NRCWF,0,NRCWF,NRCWF,SVHGA,UI,VI)
C      CALL CSVC (XIPGH,10,10,NROWC,NROWC,0,NROWC,NROWC,SVGHA,UO,VO)
C      CALL CSVC (XIPGH,10,10,NRCWF,NRCWF,0,NRCWF,NRCWF,SVHGM,UM,I,VM,I)
C      CALL CSVC (XIPGH,10,10,NROWC,NROWC,0,NROWC,NROWC,SVGHM,UMC,VMC)
C      CMEGA(NC)=W
C      SVFG1(NC)=SVHGA(NRCWF)
C      SVFG2(NC)=SVHGA(NRCWF-1)
C      SVHGX(NC)=SVHGA(1)
C      SVGH1(NC)=SVGHA(NRCWC)
C      SVGH2(NC)=SVGHA(NRCWC-1)
C      SVGHX(NC)=SVGHA(1)
C      SVFGM1(NC)=SVHGM(NROWF)
C      SVFGM2(NC)=SVHGM(NRCWF-1)
C      SVFGMX(NC)=SVHGM(1)
C      SVGFM1(NC)=SVGHM(NRCWC)
C      SVGHM2(NC)=SVGHM(NRCWC-1)
C      SVGHMX(NC)=SVGHM(1)
C      PHGAD(NC)=SCRT(SVHG1(NC)*SVHG2(NC))
C      PGHAD(NC)=SCRT(SVGH1(NC)*SVGH2(NC))
C      PHGMU(NC)=SCRT(SVHGM1(NC)*SVHGM2(NC))
C      PGHMU(NC)=SCRT(SVGHM1(NC)*SVGHM2(NC))
C      SMINAI=(SMINAI+(AMAXI(0.0,SVMINI-SVGH1(NC)))*.2)
C      SMINAC=(SMINAI+(AMAXI(0.0,SVMINO-SVGH1(NC)))*.2)
C
C      COMPUTE LCCP FOR ALL DESIRED FREQUENCY BANDWIDTH
C
C      W=W+DEL
C      WD=10.*C*WII
C      IF (W.LT.WC) GO TO 620
C      DEL=10.*C*DEL
C      WII=10.*C*WII
C      NC=NC+1
C      IF (W.LE.WMAX) GO TO 430
C      MAXN=NC-1
C
C      FCRM QUANTITIES TO USE IN THE COST CRITERIA OF OPTIMIZER
C
C      SVMIN=SVFG1(1)
C      DO 630 J=1,MAXN
C      IF (SVGF1(J).GE.SVMIN) GO TO 630
C      SVMIN=SVGH1(J)
C      CONTINUE
C
C      630
C
C      THE CEJECTIVE FUNCTION IS INSERTED HERE

```



```

12 CONTINUE
DO 30 J=1,NCA
30 I(J,J)=1.0
DO 31 J=1,NRA
DO 32 K=1,NCA
32 SI(J,J)=S*I(J,J)
31 CONTINUE
C CALL CWFITE(SI,NRA,NCA)
DO 50 J=1,NRA
DO 40 K=1,NCA
40 SIA(J,K)=SI(J,K)-AX(J,K)
50 CONTINUE
C CALL CWFITE(SIA,NRA,NCA)
CALL CFLXCV(FKCX,FKC,NRA,NCA)
CALL CFLXCV(BFX,BF,NRA,NCA)
DO 60 J=1,NRA
DO 70 K=1,NCA
70 AALG(J,K)=SIA(J,K)+FKCX(J,K)+BFX(J,K)
C CALL CWFITE(AAUG,NRA,NCA)
60 CONTINUE
CALL LEGTIC(AAUG,NCA,IO,I,NCA,IO,O,IA,IER)
CALL CPMATML(I,FKX,NRA,NCA,NCK,AAUK)
CALL CPMATML(FX,AAUK,NRF,NCF,NCK,H)
CALL CWFITE(H,NRF,NCK)
RETURN
END
C

```

```

CCN07690
CCN07700
CCN07710
CCN07720
CCN07730
CCN07740
CCN07750
CCN07760
CCN07770
CCN07780
CCN07790
CCN07800
CCN07810
CCN07820
CCN07830
CCN07840
CCN07850
CCN07860
CCN07870
CCN07880
CCN07890
CCN07900
CCN07910
CCN07920
CCN07930
CCN07940
CCN07950
CCN07960
CCN07970
CCN07980
CCN07990
CCN08000

```


LIST OF REFERENCES

1. Rosenbrock, H. H., Computer-Aided Control Systems Design, Academic Press, 1974.
2. Sandell, N. R., Jr., Recent Developments in the Robustness Theory of Multivariable Systems, Office of Naval Research Report CR-275-277-1F, 1979.
3. Kalman, R. E., "When Is A Linear Control System Optimal?", ASME J. Basic Engr., pp51-60, March 1964.
4. Anderson, B. D. C. and Moore, J. B., Linear Optimal Control, Prentice Hall, Inc., 1971.
5. Rosenbrock, H. E., "Design of Multivariable Control Systems Using the Inverse Nyquist Array", Proc. of IEEE, Vol 116, pp 1929-1936, Nov 1969.
6. Doyle, J. C. and Stein, G., "Multivariable Feedback Design: Concepts for a Classical/Modern Synthesis", IEEE Trans. Auto. Control, Vol AC-26, No.1, pp4-16, Feb 1981.
7. Safanov, M. G. and Athans, M., "Gain and Phase Margin of Multiloop LQG Regulators", IEEE Trans. Auto. Control, April 1977.
8. Sandell, N. S., Jr., Lehtomaki, N. A., and Athans, M., "Robustness Results in Linear Quadratic Gaussian Based Multivariable Control Designs", IEEE Trans. Auto. Control, Vol 26, No. 1, pp75-92, Feb. 1981.
9. Lehtomaki, N. A., Practical Robustness Measures in Multivariable Control System Analysis, Ph.D. Thesis, Massachusetts Institute of Technology, May 1981.
10. Vanderplaats, G. N., ADS - A Fortran Program for Automated Design Synthesis, Version 1.0, Naval Postgraduate School, Monterey, Ca., May 1984.
11. Vanderplaats, G. N., Numerical Optimization Techniques for Engineering Design: With Applications, McGraw-Hill, 1984.
12. Vanderplaats, G. N., CONMIN-A Fortran Program for Constrained Function Minimization - User Manual, NASA TM X-622282, AUG 1973.

13. Shapiro, E. Y., et. al., " Pole Placement with Output Feedback", AIAA J. of Guidance and Control, Vol 4, No 4, pp 441-443, July-Aug 1981.
14. Mukhopadhyay, V. and Newsom, J., Application of Matrix Singular Value Margins of Multiloop Systems, NASA TM-84524, July 1982.
15. Mukhopadhyay, V. and Newsom, J., " The use of Singular Value Gradients and Optimization Techniques to Design Robust Controllers for Multiloop Systems", AIAA Guidance and Control Papers, August 1983.
16. Sandell, N. R., Jr., et al, Multivariable Stability Margins for Vehicle Flight Control Systems, TR-121, Alphatech Inc., Dec. 1981.
17. Doyle, J. C. and Stein, G., " Robustness with Observers", IEEE Trans. Auto. Control, Vol AC-24, pp 607-611, Aug. 1979.

INITIAL DISTRIBUTION LIST

	No. Copies
1. Defense Technical Information Center Cameron Station Alexandria, Virginia 22314	2
2. Professor D. J. Collins, Code 67Co Department of Aeronautics Naval Postgraduate School Monterey, California 93943	5
3. Library, Code 0142 Naval Postgraduate School Monterey, California 93943	2
4. Department Chairman, Code 67 Department of Aeronautics Naval Postgraduate School Monterey, California 93943	1
5. Prof M.F. Platzer, Code 67F1 Department of Aeronautics Naval Postgraduate School Monterey, California 93943	1
6. Prof T.H. Gawain, Code 67Gn Department of Aeronautics Naval Postgraduate School Monterey, California 93943	1
7. Prof R.D. Strum, Code 62St Department of Electrical Engineering Naval Postgraduate School Monterey, California 93943	1
8. Prof G.E. Latta, Code 54Lz Department of Mathematics Naval Postgraduate School Monterey, California 93943	1
9. Prof G.N. Vanderplats Department of Mechanical and Environmental Engineering University of California Santa Barbara, California 93106	1
9. Commander Vernon C. Gordon, USN 315 Alameda Blvd. Coronado, California 92118	3

21 JUL 91

38216

208097

Thesis

G5762

Gordon

c.1

Utilization of numerical optimization techniques in the design of robust multi-input, multi-output control systems.

8 OCT 91

208097

208097

Thesis

G5762

Gordon

c.1

Utilization of numerical optimization techniques in the design of robust multi-input, multi-output control systems.



thesG5762
Utilization of numerical optimization te



3 2768 002 13138 5
DUDLEY KNOX LIBRARY